A NOVEL AUGMENTED REALITY BASED SYSTEM FOR PROVIDING MAINTENANCE ASSISTANCE

ZHU JIANG

(B.Eng., Nanjing University of Aeronautics and Astronautics, P.R. China)

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

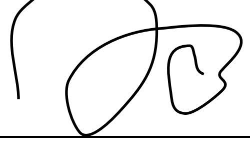
NUS GRADUATE SCHOOL FOR INTEGRATIVE SCIENCES AND ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE

2013

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.



Zhu Jiang

2013

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my supervisors, Professor Andrew Nee Yeh Ching and Associate Professor Ong Soh Khim, for their constructive suggestions on the research and constant support during my candidature. They encourage me to be confident and move forward during tough times, and instruct me to be patient and down to earth when I seek shortcuts. Their invaluable guidance has been indispensable for my PhD work.

I would like to express my gratitude to the researchers and PhD students in the Augmented Reality and Assistive Technology Laboratory, National University of Singapore. They are Dr. Zhang Jie, Dr. Shen Yan, Dr. Fang Hongchao, Dr. Wang Zhenbiao, Jiang Shuai, Ng Laixing, Andrew Yew, Yu Lu, Wang Xin, Yan Shijun, Yang Shanshan, Zhao Mengyu, and Huang Jiming. They have been great colleagues that assist my work, and best friends that enrich my life in Singapore. My special thanks to Mr. Tan Choon Huat in the Advanced Manufacturing Laboratory of National University of Singapore for helping me to conduct several case and user studies.

I would also like to thank NUS Graduate School for Integrative Sciences and Engineering, National University of Singapore, for providing me with the research scholarship.

Last but not least, I am most grateful to my family, who has been sharing the joys of my successes and always giving me a hand the time I need help. All the achievements I have made would have been impossible without your support.

TABLE OF CONTENTS

SUMMARY

Maintenance plays an important role in the equipment life cycle as it restores and improves the equipment performance, reliability, and safety. However, the increasing complexity and the technology advancement of equipment pose constant challenges to the maintenance technicians nowadays. Augmented reality (AR) technology, which is an interface between the virtual and real world, can be used to enhance maintenance activities. AR is an enabling technology for man-machine-information interaction that can assist maintenance activities. Using AR, maintenance information in multi-media forms can be augmented and aligned virtually in the maintenance area, and collaborative work between the maintenance technicians and the remote experts and engineers can be enabled and enhanced through visual interactions. Thus, AR is a better approach for providing maintenance information as compared to paper prints and computers and can potentially improve the workflow of remote maintenance.

In this research, the reported AR-assisted maintenance systems have been analyzed and the opportunities of applying AR technology to routine and *ad hoc* equipment maintenance activities have been explored. Specifically, a novel AR-assisted maintenance system (ARAMS) has been proposed to assist the technicians in their daily maintenance activities. ARAMS has two features, i.e., providing a remote collaboration mechanism that allows the experts to create and provide AR-based visual instructions to the technicians efficiently and effectively, and exploring the authoring and visualization of context-aware AR contents.

Using the ARAMS system, context-relevant information can be provided to the technicians, i.e., the information will be provided according to the status of the maintenance activity and the technician's expertise level. In addition, the AR contents can be interacted with instead of being "read-only", so that the technicians can rectify any incorrect AR contents that have been created by the AR developers and author AR contents to record and share the accumulated knowledge and experience on equipment maintenance with other technicians. In remote

maintenance, where the technician may encounter difficult situations and seek assistance from a remote expert, the ARAMS system can assist the remote expert to instruct the technician through authoring AR-based instructions online conveniently.

To achieve these functions of the proposed system, authoring and the generation of AR contents are addressed in this research. A bi-directional authoring tool, which enables the AR developers to create context-relevant information via a desktop 2D user interface and the maintenance technicians to author AR contents on-site, is proposed. In addition, an online authoring tool is proposed to enable an expert to create intuitive AR-based instructions in an environment with no *a priori* knowledge conveniently and efficiently. In this prototype, a Dual Window User Interface (DWUI) is proposed to broaden the expert's view and improve the robustness of the authoring process. Based on these proposed methodologies, a prototype of the ARAMS system has been developed, and three user studies have been conducted to validate the proposed concepts and techniques and test the usability of the ARAMS system. The user studies indicate that the proposed system can assist the AR developers to author context-aware AR contents effectively, the authored contents can be adapted to the contexts correctly, and the provided AR information can improve the maintenance work flow as compared to paper manuals. In remote maintenance, ARAMS enables more efficient and less error prone remote collaboration. Finally, user feedback has been collected to provide suggestions on further improvement of the prototype system in future.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AR	Augmented reality
API	Application Programming Interface
APRIL	Augmented Presentation and Interaction Language
ARAMS	AR-Assisted Maintenance System
ARV	AR-based Visualization
CM	Context Management
COARE	Context Ontology for AR Environments
CV	Computer Vision
DART	Designer’s Augmented Reality Toolkit
DWUI	Dual Window User Interface
GUI	Graphical User interface
HHD	Hand-Held Display
HMD	Head-Mounted Display
KARMA	Knowledge Based Augmented Reality
LOD	Level of Detail
MARS	Mobile Augmented Reality System
OC	Opportunistic Controls
OFA	OFfline Authoring
ONA	ONline Authoring
OSA	On-Site Authoring
PTAM	Parallel Tracking and Mapping
RDF	Resource Description Framework
RFID	Radio Frequency Identification
RV	Reality-Virtuality
SLAM	Simultaneous Localization And Mapping
SWRL	Semantic Web Rule Language
VR	Virtual Reality
VPD	Virtual Pointing Device
WYSIWYG	What You See Is What You Get

Chapter 1 Introduction

Maintenance plays an important role in the equipment life cycle as it restores and improves the equipment performance, reliability, and safety. However, the increasing complexity and the technology advancement of equipment pose challenges to the maintenance technicians nowadays. Augmented reality (AR) technology, which is an interface between the virtual and real world, can be used to enhance maintenance activities. AR technology is an enabling technology for man-machine-information interaction that can assist maintenance activities. Using AR, maintenance information in multi-media forms can be augmented and aligned virtually in the maintenance area, and collaborative work between the maintenance technicians and the remote experts and engineers can be enabled and enhanced through visual interactions. Thus, AR provides a better approach for providing maintenance information as compared to paper prints and computers and can potentially improve the workflow of remote maintenance.

1.1 Equipment Maintenance

Equipment maintenance activities can be classified into two categories, namely, preventive maintenance and corrective maintenance. Preventive maintenance is conducted to keep the equipment working and/or extend the life of the equipment, and corrective maintenance is carried out to fix mal-functioning equipment. To conduct maintenance activities, the technicians usually have to refer to the paper manuals, mobile devices, or computers for the maintenance information, e.g., schedules for preventive maintenance activities and instructions on the corrective maintenance activities. For example, the maintenance procedures can be referred from the hypermedia maintenance manual developed by Setchi and White [2003] (Figure 1.1). However, such information reference processes are physically and psychologically stressful for the technician. The technician is required to switch his context back and forth from the equipment maintenance area to the references. In addition, the technician has to filter the useful information from the references and map this information to the real equipment. Thus, it decreases the work efficiency and can inject errors to the maintenance work.

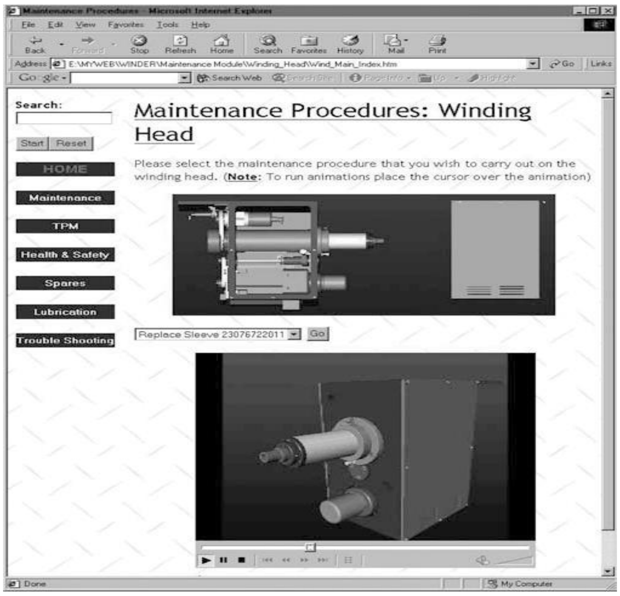


Figure 1.1: Hypermedia maintenance manuals [Setchi and White 2003]

In the case of difficult situations where the technician needs assistance from an expert who is located remotely, telephone and email are usually used for remote collaboration. Using these tools, it is inconvenient for the technician to describe the situations to the expert and unintuitive for the expert to present the maintenance instructions to the technician. As the collaborative work relies heavily on verbal and textual communications, visual interactions are barely supported. Thus, the tools supporting remote maintenance need to be enhanced with more intuitive visual interactions.

1.2 Augmented Reality

AR is a recently developed technology that allows the users to perceive the environment with additional information that has been registered spatially and temporally with the real scene [Azuma 1997]. AR allows the users to perceive and interact with both real and virtual objects instead of immersing the users in artificial reality as VR (Virtual Reality). Milgram and Kishino [1994] proposed the reality-virtuality continuum scheme (Figure 1.2) to show the relationship between AR and VR.

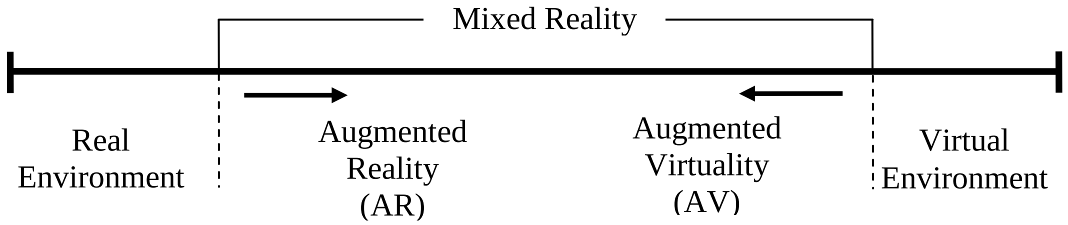


Figure 1.2: Reality-Virtuality (RV) Continuum [Milgram and Kishino 1994]

There are three characteristic properties of AR systems, namely, (1) combining virtual and real objects in a real environment, (2) running interactively and in real time, and (3) registering real and virtual objects with each other [Azuma 1997].

AR technology can be used to assist the maintenance technicians through augmenting maintenance information virtually in the maintenance area. For example, maintenance instructions can be presented using computer graphics and aligned with the real printer (Figure 1.3) [Feiner *et al.* 1993]. Thus, AR technology provides a better approach for providing maintenance information to the technicians as compared to paper prints and computers, as it can align the information with the real equipment virtually and improve information presentation using multi-media files.

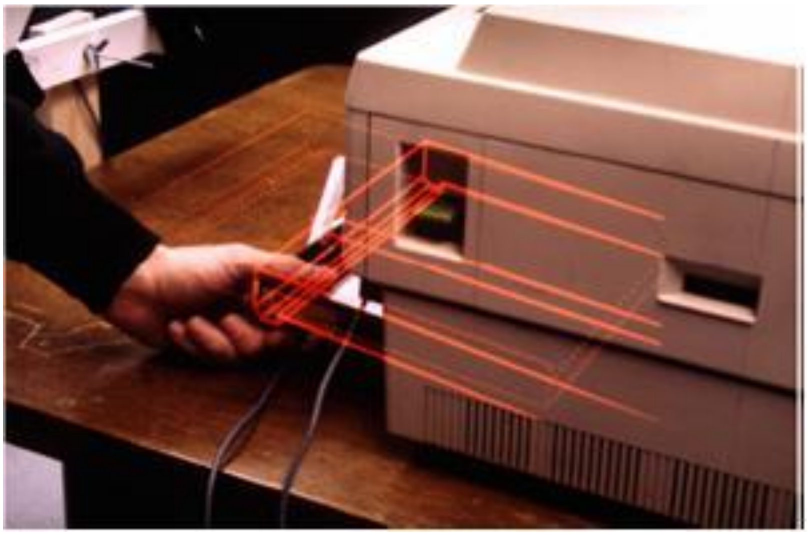


Figure 1.3: Knowledge-Based Augmented Reality (KARMA) [Feiner *et al.* 1993]

AR technology can also be used to enhance the collaborative work between the maintenance technician and the remote expert through enabling visual interactions. For example, the expert can instruct the technician to replace a specific disk by highlighting the disk using a rectangle and displaying the action, and replacing the disk on top (Figure 1.4) [Reitmayr *et al.* 2007]. Thus, AR technology can potentially improve the workflow of remote maintenance.

1.3 Research Motivations and Objectives

Although significant efforts have been devoted to applying the AR technology to equipment maintenance, no single AR system has yet been proven to be well accepted by the industry [Nee *et al.* 2012]. This research is to identify the open research issues through reviewing the reported AR-assisted maintenance systems, and improve the system usability by addressing certain issues that have been identified.

Figure 1.4: Remote collaboration using AR technology [Reitmayr *et al.* 2007]

1.3.1 Context-aware maintenance information

Despite the promising features of rendering maintenance information using AR technology, to improve the usability of the state-of-the-art AR maintenance systems in routine and *ad hoc* maintenance activities, the virtual information provided should be adapted to the various contexts, i.e., context relevant. Context is any information that can be utilized to characterize the situation of an entity, where the entity can be a place, a person, and/or an object that is relevant to the interaction between a user and an application, such as time, location, and activities [Dey 2001]. A system can be deemed as context-aware if it can collect, reason, and utilize context information, and adapt its functionality to the varying contexts [Byun and Cheverst 2004]. For example, the operator's location can be used as the context to infer the equipment information that the technician may be interested in. When the technician is in the area of a certain equipment, the information of this equipment will be deemed as useful and can be rendered to him.

In addition, the rendered AR contents should not be "read-only", such that the maintenance technicians are merely passive information receivers. Instead, the technicians should be able to interact with and edit the AR contents. Firstly, providing the technicians with the means to create AR contents on-site is useful as it allows them to record and share the accumulated knowledge and experience on equipment maintenance. For example, a newly detected fault that can result in certain equipment failure should be added into the maintenance knowledge base so that it can be used by the AR-assisted maintenance systems to alert other technicians. Secondly, providing the technicians with the means to edit the AR contents is useful for correcting any inappropriate AR contents that may have been created by the AR developers. As the AR developers usually cannot review the authored AR contents on-site using the state-of-the-art AR authoring systems, and the differences in the perception of the AR contents through the desktop AR authoring systems and those on-site can result in errors. However, such errors can be corrected easily by the on-site technicians. For example, the compact structure of current equipment poses challenges for the AR developers to register the virtual objects on the equipment precisely via the desktop 2D user interfaces, but it is relatively easier for the technicians to arrange the objects spatially on-site.

1.3.2 AR in remote maintenance

Although AR technology has been proven to be an effective way to enhance the collaboration of 3D tasks [Kuzuoka 1992], the opportunities of applying AR technology in remote maintenance has yet to be explored thoroughly in the reported works. To employ AR technology in remote maintenance effectively, an appropriate authoring tool is required for the experts to create the virtual instructions. Such an authoring tool should have the following features:

- Online authoring. The maintenance instructions can be created and rendered online instead of being prepared offline, so that the collaborative work can be conducted smoothly and efficiently.
- High mobility. The instructions can be authored and rendered in unprepared environments, so that remote collaboration can take place wherever the technician needs assistance.
- Stable Authoring. Without *a priori* knowledge of the maintenance environment, the live video captured by the camera worn on the technician is often the only channel where the expert can access the remote site and use as the interface for authoring virtual instructions. However, the live video can be jittery due to the unavoidable movements of the technician. Thus, a stable authoring interface should be provided to the experts.

1.3.3 Research objectives

The objectives of this research are to:

- Design and develop an AR-assisted maintenance system that will provide context-aware information to the technicians.
- Design and develop a mobile user interface that allows the technicians to interact with the virtual information rendered.
- Design and develop an AR-assisted maintenance system that allows the expert to instruct the technician using AR technology efficiently and effectively.

Specifically, the research issues to achieve the above-mentioned objectives are

- Designing and developing an authoring system for the AR developers to create context-aware information for various AR applications offline and an adaptation engine that filters and renders the authored information to the users according to the contexts online.

- Designing and developing an authorable context-aware system to assist the maintenance technicians through providing the technicians with context-aware information and allowing them to edit and update AR contents using a mobile user interface.
- Designing and developing an online authoring tool for remote maintenance, so that the expert can create AR-based instructions in unprepared environments through a stable user interface.

1.4 Research Scope

This research aims to provide context-aware information to the maintenance technicians using AR technology, and focuses on the authoring of context-aware AR contents. However, acquiring and analyzing the technicians' contexts accurately and efficiently is important for filtering the right information online, but they are not within the scope of this research and would need to be addressed in future.

In addition, the proposed remote maintenance system is designed for the maintenance technicians to seek assistance from remote experts, where the experts can provide instructions using AR technology. The system mainly focuses on the collaborative work on equipment maintenance, and the potential of applying the system to other domains will be explored in future.

1.5 Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 presents a review of the state-of-the-art AR maintenance systems with respect to display technologies, tracking technologies, human-system interaction, data management, and remote collaboration, and the research issues are discussed. The reported works on integrating AR technology with context-awareness are discussed. In addition, the state-of-the-art AR authoring systems have been analyzed, and the potential research directions have been identified.

Chapter 3 describes the proposed AR-assisted maintenance system (ARAMS) to assist the maintenance technicians in daily maintenance activities. The overall system architecture is presented, the functionality of each module is illustrated, and the research tasks involved to achieve the proposed system are discussed.

Chapter 4 describes the proposed online authoring tool to achieve the remote collaboration function of the ARAMS system. The methodologies of the authoring tool and remote collaboration scheme are described.

Chapter 5 describes the proposed concepts and techniques of context-aware AR contents authoring and visualization. The bi-directional authoring tool that allows the AR developers to create context-aware information offline and the maintenance technicians to edit the provided information on-site is presented. In addition, the filtering and adaptation of the authored information to provide context-aware AR services is described.

Chapter 6 describes the system implementation and the user studies conducted to validate the proposed methodologies, evaluate the system performance, and collect user feedback.

Chapter 7 concludes the research, highlights the contributions, and presents several recommendations for future work.

Chapter 2 Research Background

In this chapter, the reported works on applying AR technology to equipment maintenance are reviewed, and the issues of these reported AR-assisted maintenance systems are presented. In addition, the state-of-the-art of integrating AR technology with context-awareness and AR authoring systems are analyzed and the open research issues are identified.

2.1 AR-assisted Maintenance System

The application of AR technology in aiding the maintenance technicians started in the early 1990s, and the objectives of the reported work have changed from demonstrating the benefits of applying AR to maintenance to improving the usability of such research-based AR systems in routine and *ad hoc* maintenance activities.

As one of the pioneering AR maintenance systems, knowledge-based augmented reality for maintenance assistance (KARMAR) [Feiner *et al.* 1993] was designed primarily for providing maintenance instructions to the users. The maintenance database, which consists of maintenance instructions, is developed in advance and considered as maintenance "knowledge". With sensors pre-installed on a printer, KARMAR is able to estimate the relative position of the user's head to the printer, retrieve the corresponding instructions and augment the information in the user's view through a Head-Mounted Display (HMD) (Figure 2.1). Although rather rudimentary, this prototype system has validated the benefits that AR can bring to equipment maintenance.

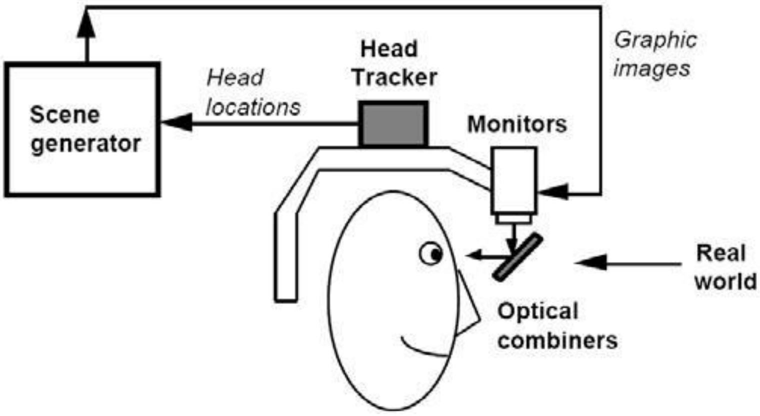
Figure 2.1: KAMAR system setup [Feiner *et al.* 1993]

To improve the usability of current research-based AR maintenance systems in real maintenance scenarios, a few issues have been worked on by several research institutes since the mid 1990s [Feiner *et al.* 1993, Harmo *et al*. 2001, Friedrich 2002, Schwald *et al*. 2001, Didier and Roussel 2005, ARTESAS, Savioja *et al*. 2007, Henderson and Feiner 2009, Lee and Akin 2011, Fantini *et al.* 2011]. These reported systems will be reviewed according to five important issues of implementing AR-assisted maintenance systems, namely, display technologies, tracking technologies, data management, human-system interaction, and remote collaboration (Table 2.1).

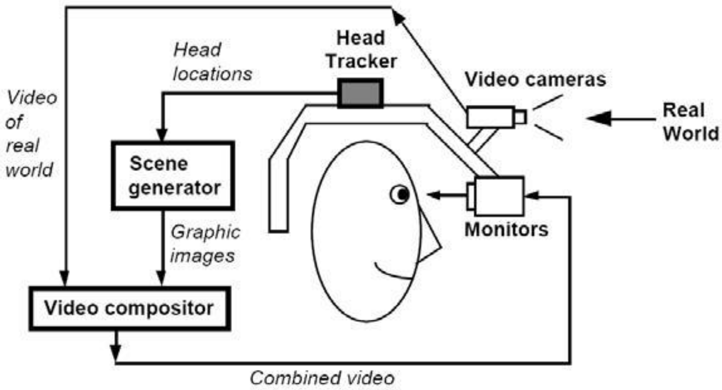
2.1.1 Display technologies

Display technologies, which overlay real and virtual objects, play an important role in AR applications. They generally fall into three categories, namely, head-mounted, hand-held and projective display technologies.

Head Mounted Displays (HMDs) present an augmented environment to either one eye (*monocular*) or two eyes (*binocular*) of the users. HMDs can be categorized into optical see-through displays and video see-through displays according to the way they overlay the real and virtual environments [Milgram *et al.* 1994]. Optical see-through displays provide a direct view of the real world to the user, and overlay virtual objects on top of the real scene (Figure 2.2(a)). Video see-through displays combine the real scene captured by cameras and virtual objects digitally or via video-mixing hardware, and present the augmented scene on the display (Figure 2.2(b)).



(a) Optical see-through architecture,



(b) Video see-through architecture

Figure 2.2: Optical Vs. Video [Azuma 1997]

Hand-Held Displays (HHD), such as Tablet PCs, mobile phones and portable game machines, feature one or more cameras and an LCD screen. As video see-through displays, HHDs overlay virtual content on captured images of the real scene to create an experience referred to as the magnifying glass [Rekimoto 1995].

Projective displays project virtual contents directly onto the real world [Bimber and Raskar 2005]. Thus, they free the users from wearing a display or carrying a computer. However, they typically assume that all the virtual contents should lie on the same projected surface, and this subsequently limits the types of geometry that can be presented. Thus, projective displays are not as useful as HMDs and HHDs with regards to applying AR to maintenance tasks.

HMDs [Feiner *et al*. 1993, Friedrich 2002, Schwald *et al*. 2001, ARTESAS, Fantini *et al.* 2011] and handheld devices [Savioja *et al*. 2007, Didier and Roussel 2005] are the prevalent display devices employed in the reported systems. However, it has been reported that these maintenance operators do not like to wear any type of HMD [Didier and Roussel 2005]. In addition, HHDs require the user to have at least one hand to hold and interact with the devices. It is inconvenient as the user's hands are usually engaged in maintenance tasks. Thus, a few alternative display devices, such as chest worn devices [Sakata *et al.* 2006], have been proposed, but well-accepted display devices are not available yet.

2.1.2 Tracking

Tracking is a fundamental issue in AR, and it is the process of determining the position and orientation of a real object. Accurate and stable tracking techniques are especially important for applying AR in maintenance applications as equipment maintenance requires a high level of skill and precision.

Tracking technologies generally fall into three categories, namely, sensor-based (e.g., mechanical, inertial, ultrasonic and electromagnetic sensors), optical and hybrid. Hybrid tracking employs two or more tracking methods, and fuses the data to produce a single estimate of the position and/or orientation. It supplements the weaknesses of one tracking technology with the strengths of the others. Sun *et al.* [2007] proposed a hybrid tracking system that utilizes data from the GPS receiver and inertial sensors to estimate the position and orientation of the user, and precise tracking can be realized by matching images captured using the camera and those stored in the database. With the fast development of Computer Vision (CV) technologies, optical tracking has become dominant in maintenance applications due to its low cost and hardware simplicity (Table 2-1).

Optical tracking can be further categorized into marker-based tracking and markerless tracking. Marker-based tracking uses cameras to detect predefined markers attached at known locations in the environment in advance. ARToolKit [ARToolKit 2005] and ARTag [Fiala 2005] are two well-known marker-based tracking methods. These tracking methods have been adopted in many AR systems [Friedrich 2002, Schwald *et al*. 2001, Savioja *et al*. 2007] as they are able to provide a relatively high level of accuracy and the required software and hardware have become common items. Using ARToolKit, the position and orientation of the camera relative to the fiducial markers attached in the environment will be estimated using CV technologies, and the virtual media files can be registered in the real environments. The working process of ARToolKit [ARToolKit 2005] consists of four steps: (i) transforming a live video frame into a binary image; (ii) searching squares in the binary image; (iii) matching every detected square with the pre-trained patterns; and (iv) determining the position and orientation of the camera relative to the marker if a detected square matches with a pre-defined pattern (Figure 2.3).

However, marker-based tracking methods require the environment to be prepared in advance and the markers to be completely visible during the tracking process. These limitations prompted researchers to develop markerless tracking methods.

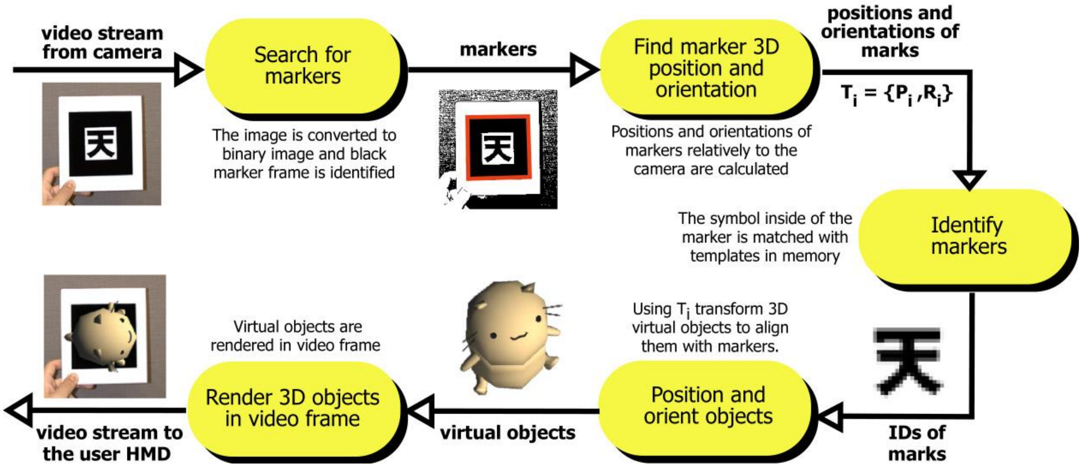


Figure 2.3: Diagram of ARToolKit [ARToolKit 2005]

Fantini *et al.* [2011] utilized a reference-image-based approach for tracking. Reference images of the maintenance environment are captured and their local invariant features are extracted using CV technology in advance (Figure 2.4). During the online process, each incoming frame captured by the camera will be analyzed and its local invariant features will be matched with those of the reference images to estimate the camera poses in real time. However, this approach assumes that the camera is viewing a planar object, which is not generic for various maintenance environments.

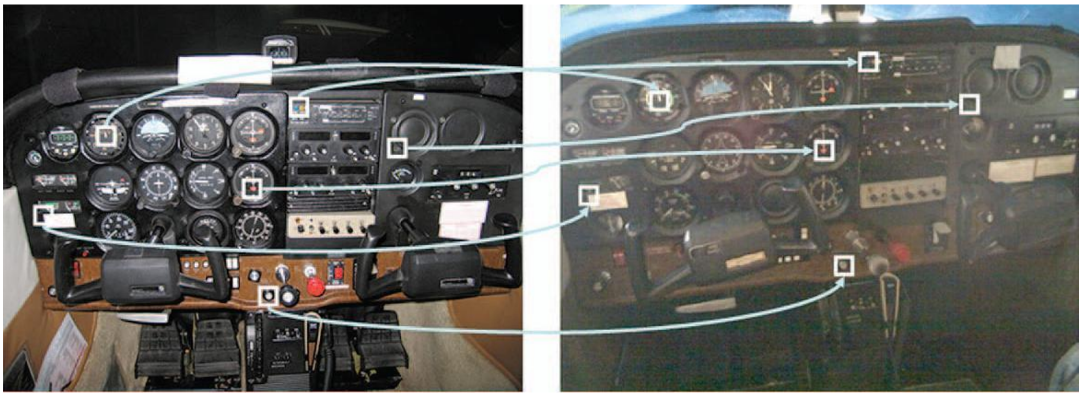


Figure 2.4: Feature matching [Fantini *et al.* 2011]

3D model-based tracking methods that match the point and edge features extracted from each frame captured online with those extracted from prebuilt CAD models have been proposed [ARTESAS, Alvarez *et al*. 2011], These methods remove the need to attach markers in the maintenance area, but the CAD models of equipment and components are usually unavailable *a priori* and it could be costly to build them in advance. Neubert *et al.* [2012] proposed a solution which allows rapid generation of rough, appearance-based edge models comprising groups of planes (Figure 2.5). A handheld device with a camera is provided to capture a video of the object of interest, and a list of edges can be annotated on a set of keyframes. The generated models will be used for estimating the camera poses online through edge matching. Caponio *et al.* [2011] proposed a similar approach that analyzes the video feed of the application environment to extract and label the feature points offline, which will be used to estimate the position and pose of the objects online.

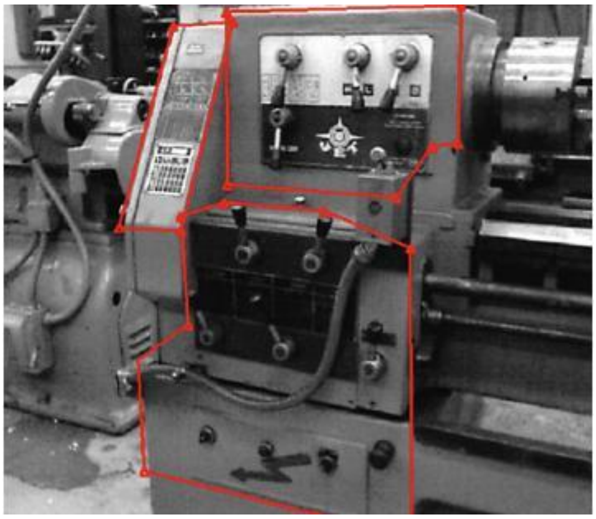


Figure 2.5: The lathe model created [Neubert et al. 2012]

SLAM (Simultaneous Localization and Mapping) technique removes the offline labeling process, and allows simultaneous map construction and tracking online. Klein and Murray [2007] presented the PTAM (Parallel Tracking And Mapping) system that is able to construct a 3D map consisting of feature points and keyframes of unknown environments. The camera pose can be estimated through analyzing the feature points in each incoming frame and match the features with those in the 3D map. Yan *et al.* (2009) explored vision-based SLAM using line features as line features are abundant in nuclear power plants. In the proposed method, Plücker coordinates are used to represent the 3D lines, and Gaussian sum is used to approximate the initial state of the features and update the new observations. Such tracking methods require limited offline preparation of the application scenario and provide competitive tracking performance, and they are the current research direction.

2.1.3 Data management

Allowing the maintenance technicians to retrieve pertinent information, e.g., maintenance instructions and equipment maintenance history, efficiently is one of the most important benefits AR can bring to maintenance applications.

The virtual model-based data management method was proposed in Etälä [Harmo *et al.* 2001]. This system requires the technician to first align the virtual model of equipment with the real counterpart, and this initial step bridges the gap between the real and virtual worlds (Figure 2.6). Subsequently, the technician is able to retrieve and update the information from the database by interacting with the real components. For example, the technician could look up certain information of a particular component by pointing at it, while the system identifies its virtual counterpart and retrieves the related information from the database at the same time. This virtual model-based method improves the efficiency and accuracy of on-site data management. However, virtual models of equipment are usually not available, and it is inconvenient for the user to overlay them precisely onto their real counterparts.

The scenario/process-oriented data management method is more prevalent among the reported AR maintenance systems [Feiner *et al.* 1993, Friedrich 2002, Schwald *et al.* 2001, ARTESAS, Henderson and Feiner 2009]. These systems predefine the information to be rendered to the technician in a particular scenario/process, and render the information when the scenario/process has been identified. For example, the instructions on a particular maintenance step will be provided to the technicians when this step is being conducted.

Figure 2.6: Overlay the virtual model on the real equipment [Harmo *et al.* 2001]

This review indicates that context-awareness has not been thoroughly explored in the reported systems, and the information provided is not customized according to the specific needs of the technicians. For example, the technicians with different levels of expertise may require different instructions on a particular maintenance step. In addition, the systems barely allow the technicians to interact with and update the information provided, i.e., being authorable, and the technicians can only receive the provided information passively.

One of the challenges of developing such authorable context-aware AR systems is that there is not yet an appropriate authoring tool that can enable the AR developers, as well as the on-site technicians to create context-aware contents.

2.1.4 Human–system interaction

Human–system interaction is another issue that hinders current AR maintenance systems in routine and *ad hoc* maintenance activities. A commonly acceptable interaction approach is not available yet although many useful techniques and tools have been reported [Goose *et al.* 2003, Park *et al.* 2008, Henderson and Feiner 2010]. The prevalent human–system interaction tools, such as mouse, keyboard, and touch screen, have been employed and reported in several systems [Savioja *et al*. 2007, Didier and Roussel 2005]. These tools require less user training, but they usually distract the users from the maintenance operations as the users have to halt the maintenance work temporarily and use the interaction tools to interact with the systems.

Recently, several new human-system interaction approaches have been proposed [Schwald *et al.* 2001, Goose *et al.* 2003, Park *et al.* 2008]. STARMATE [Schwald *et al.* 2001] is a Virtual Pointing Device (VPD) for the users to select items in the virtual and real worlds. As shown in Figure 2-7 (a), the VPD is a ray, with origin between the user's eyes, pointing into the selectable objects. Thus, the user could interact with the system via head movements. Goose *et al.* [2003] developed a speech-enabled AR framework for mobile maintenance. The system allows the user to "talk" to the equipment as if he is talking to another person. Their system identifies the user's location via the markers positioned in the worksite in advance, and triggers a context-specific speech dialog. However, speech-based interaction should preferably be kept to a minimum as the application environment of AR maintenance systems, such as factories, are typically very noisy. Eye gazing-based interaction has been proposed in the wearable AR system reported by Park *et al.* [2008]. By equipping a see-through HMD with an eye tracker, it is able to track the user's eye movements for interaction with the system (Figure 2-7(c)). However, several challenges, such as the "Midas Touch" problem [Jacob 1993], have to be solved before gaze-based interaction can be widely adopted.

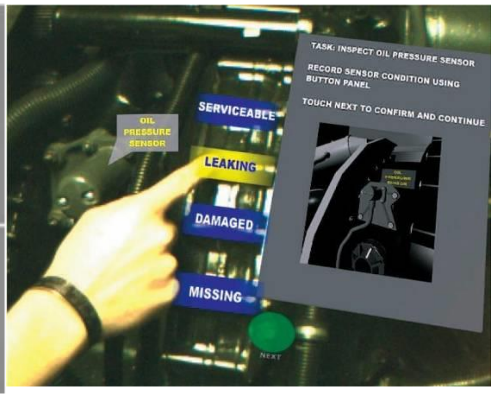
Henderson *et al.* [2010] proposed the opportunistic controls (OC) techniques, which were designed based on two criteria, namely, (1) limit extraneous hand, head, and eye movements beyond the user's immediate vicinity, and (2) provide passive haptic feedback without modifying the application environment. An OC interface aims to enable user interaction with the system through touching natural surfaces within the task environment. For example, the system renders a virtual button on the raised geometry of an engine, and a user can press the button while receiving passive haptic feedback from the raised geometry underneath (Figure 2-7 (b)).

2.1.5 Remote collaboration

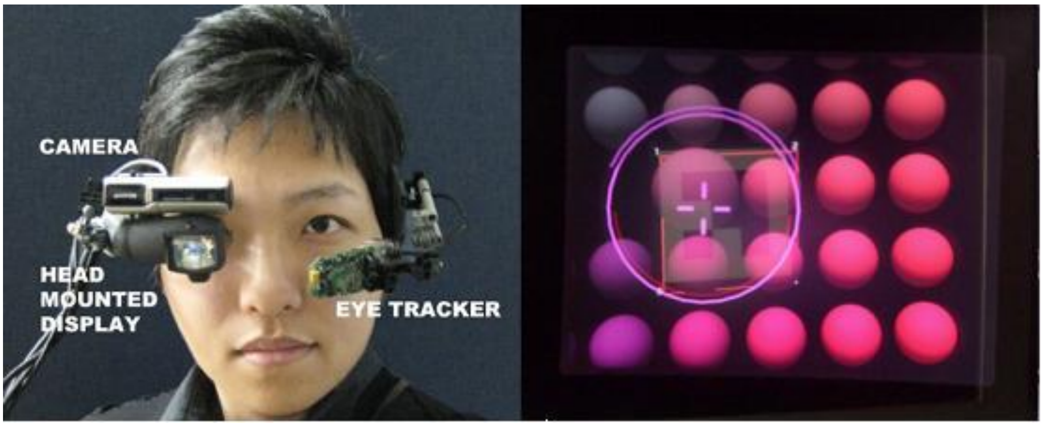
AR technology can bring promising benefits to collaborative work, but few AR-assisted systems can support remote maintenance (Table 2.1).

In remote maintenance, the role of the office-based users is to provide data, information and maintenance instructions to the on-site technician when needed. Thus, the system should support effective and efficient communication between the office-based users and the on-site maintenance technicians. Verbal instruction is most common in collaborative systems, but certain maintenance operations cannot be described easily using speech alone. Additional hardware has been employed in a few systems to facilitate tele-assistance. For example, in [Harmo *et al.* 2001, Friedrich 2002, ARTESAS], a laser pointer is worn on the body of the on-site technician, and the office-based user can control it remotely to highlight certain components (Figure 2.8).

(a) VPD [Schwald *et al.* 2001]



(b) OC [Henderson *et al.* 2010]



(c) Eye gazing-based interaction [Park *et al.* 2008]

Figure 2.7: Human-system interaction

However, AR technology is yet to be fully utilized to enhance the visual interaction in remote maintenance. In the reported systems [Harmo *et al.* 2001, Friedrich 2002, ARTESAS, Reitmayr *et al.* 2007], only simple AR-based maintenance instructions can be created and used for instructing the technicians. This is due to the lack of AR authoring tools that can be used for the expert to create AR-based contents online efficiently.

(a) (b)

Figure 2.8: (a) Expert views the maintenance environment, and can point real world objects with a laser attached on the technician, and (b) technician using a wearable computer receives instructions from the expert. [Harmo *et al.* 2001]

2.1.6 Summary

Although significant efforts have been devoted to applying the AR technology to equipment maintenance, no single AR system has yet been proven to be well accepted by the industry [Nee *et al.* 2012]. There are several issues which need to be addressed.

1) Display devices, which are comfortable to wear and do not distract the user from maintenance tasks, are not available yet.
2) The accuracy and robustness of existing tracking and registration methods have to be improved, as maintenance tasks require a high level of precision.
3) Context-aware information should be rendered to the technicians, and appropriate on-site authoring tools should be provided for the technicians to interact with the AR contents.
4) Human-system interaction techniques for AR applications are still immature. There is a need to develop intuitive user interfaces that can enable AR application users to interact with the system via convenient and natural interactions.
5) AR-based visual interactions should be fully utilized in remote maintenance, and the enabling techniques, e.g., online authoring tools, should be improved.

This research addresses the third and fifth issues to improve the usability of the state-of-the-art AR maintenance systems in routine and *ad hoc* maintenance activities. Specifically, this research focuses on the issues of providing context-aware AR information and AR authoring. Thus, the state-of-the-art of integrating AR technology with context-awareness and AR authoring will be reviewed and analyzed in sections 2.2 and 2.3 respectively.

Table 2.1: AR-assisted maintenance systems

System	Display device	Tracking technologies	Data management	Human-system interaction		Remote collaboration
				Office-based user	On-site user	
KARMAR [Feiner *et al.* 1993]	HMD	Sensor-based	Scenario - oriented/ Process-oriented	NA	Head motion	NA
Etälä [Harmo *et al.* 2001]	HMD	Marker-based	Virtual model-based	Mouse, Keyboard	Mouse, Keyboard	Microphone, Remote laser pointer
ARVIKA [Friedrich 2002]	HMD	Marker-based/ Sensor-based/ Hybrid (marker & sensor-based)	Scenario-oriented/ Process-oriented	Mouse, Keyboard	Microphone	Microphone, Remote laser pointer
STARMATE [Schwald *et al.* 2001]	HMD	Marker-based	Scenario - oriented/ Process-oriented	NA	Microphone, Virtual Pointing Device (VPD)	NA
AMRA [Didier and Roussel 2005]	HHD	Marker-based/ Hybrid (marker & CAD model-based)	NA	NA	Touch screen	NA
ARTESAS [ARTESAS]	HMD	CAD model-based	Scenario - oriented/ Process-oriented	Mouse, Keyboard	Microphone	Microphone, Remote laser pointer
PLAMOS [Savioja *et al.* 2007]	HHD	Marker-based / WLAN	Scenario - oriented/ Process-oriented	NA	Touch screen	NA
ARMAR [Henderson and Feiner 2009]	HMD	Hybrid (marker & sensor-based)	Scenario - oriented/ Process-oriented	NA	Opportunistic Control [Henderson and Feiner 2010]	NA
AROMA-FF [Lee and Akin 2011]	HMD	Marker-based	Scenario - oriented/ Process-oriented	NA	Digital camera	NA

2.2 Context-aware AR systems

AR systems that make use of context information have been reported since the 1990s. In the earlier prototypes, users are required to wear a mobile platform, which consists of a laptop, a camera, a HMD, etc. (Figure 2.9), and collect limited low-level contexts, e.g., location information, in order that the systems can provide context-adaptable AR services to them [Bauer *et al.* 2001]. Although these works are rudimentary and do not explicitly label themselves as context-aware AR systems, they have demonstrated the importance of integrating context-awareness with AR.

Figure 2.9: Mobile platform [Bauer *et al.* 2001]

Several context-aware AR systems have been developed for various applications recently [Venezia and Marengo 2010, Lee and Rhee 2008, Kim *et al.* 2011, Aydın *et al.* 2012]. A pedestrian navigation system (Figure 2.10(a)) has been developed to recommend the points of interest according to a user's location, profile, etc. [Venezia and Marengo 2010]. A U-car service system (Figure 2.10(b)) has been proposed for car maintenance work using context adaptable 3D visualization [Lee and Rhee 2008]. An intelligent learning system (Figure 2.10(c)) has been reported to provide appropriate learning contents to the users according to their profiles [Kim *et al.* 2011]. ARCAMA-3D (Figure 2.10(d)) was developed to assist the users in discovering their surroundings through providing geographic information [Aydın *et al.* 2012]. These recent works provide more in-depth analyses of the benefits and issues of developing context-aware AR systems for particular application domains. In addition, these systems have shifted from the traditional backpack platforms to more portable mobile devices, e.g., PDA and smart phones. These ubiquitous mobile devices provide both opportunities and challenges for the development of context-aware AR systems. Their built-in sensors, e.g., camera, GPS and compass, provide useful contexts of the users, and the various communication channels, e.g., wireless, Bluetooth, and GPRS, facilitate their connection with other devices. However, these mobile devices have several limitations, e.g., lower computing power, limited battery life and small screen size. Thus, the system architecture and algorithm should be carefully designed and optimized to achieve sufficient performance on mobile devices.

(a) Pedestrian navigation system [Venezia and Marengo 2010]

(b) U-car service system [Lee and Rhee 2008]

(c) Intelligent learning system [Kim *et al.* 2011]

(d) ARCAMA-3D [Aydın *et al.* 2012]

Figure 2.10: Context-ware AR systems

The fundamental issues of integrating context-awareness with AR have been researched. A software platform for context-aware AR services, intGuide has been reported [Demiris *et al.* 2005]. This system provides useful guidelines on the software framework of context-aware AR systems, but it is not generic as limited contexts, i.e., only location and orientation, are considered. Lee *et al.* [Lee *et al.* 2008] and Zhang *et al.* [Zhang *et al.* 2010] proposed two frameworks for context-aware AR services, and their frameworks generally comprise three parts, namely, Context Collection (CC), Context Processing (CP), and AR Service (ARS). These frameworks interface with physical sensors, devices and users to acquire contexts, manage and reason the contexts to provide context-adaptable 3D visualization services to the users. CAMAR 2.0 [Shin *et al.* 2009] enhances these two frameworks by allowing users to create and share AR contents with other users.

Literature review reveals the integration of context-awareness with AR can provide promising benefits for the various application domains. However, there is a lack of supporting tools for the development of context-aware AR applications, in particular the authoring of context-aware AR contents.

2.3 Authoring for AR applications

Authoring is the generation of the contents to be rendered in AR environments, and the reported AR authoring systems are reviewed and the research issues are identified in this section.

2.3.1 AR authoring systems

Various aspects of AR, e.g., tracking, interaction, and display techniques, have been actively under development [Zhou *et al.* 2008]. However, authoring AR content is usually neglected by most researchers, and it still takes places mostly at the source code level. ARToolKit [ARToolKit 2005] and OSGART [Looser *et al.* 2006] are the two low-level authoring tools that provide modular programming frameworks, and the users can create AR contents by extending the programs. However, programming skills are required to use these tools, and this hinders domain experts who are not programmers, e.g., designers, artists, and engineers, to participate in the authoring process.

To relieve the users from low-level programming, a mark-up language, APRIL (Augmented Presentation and Interaction Language), has been reported [Ledermann and Schmalstieg 2005]. APRIL abstracts the low-level programming techniques and exposes the high-level concepts, that describe the hardware setup (tracking devices and displays), the AR content and its temporal structure and interactive capabilities, to the users. However, it is still preliminary as no intuitive Graphical User interface (GUI) is provided for the users to visualize the creation of AR applications. Several comprehensive AR authoring systems, which improve their usability through various approaches, have been reported [Haringer and Regenbrecht 2002, MacIntyre *et al.* 2004, Grimm *et al.* 2002, Knopfle *et al.* 2005, Güven and Feiner 2003, Lee and Kim 2009, Lee *et al.* 2011, Reitmayr *et al.* 2007].

PowerSpace [Haringer and Regenbrecht 2002] and DART (Designer's Augmented Reality Toolkit) [MacIntyre *et al.* 2004] make use of existing authoring tools, as utilizing the available concepts and functions of the existing tools can shorten the system development time and make the users, who are already familiar with the existing tools, more willing to accept and use the developed AR authoring systems. PowerSpace is built upon the Microsoft PowerPoint, and it features fast and easy generation of AR worlds. Using PowerSpace, authoring is divided into four main steps: (i) generate and arrange the elements, e.g., images and texts, initially in 2D using Microsoft PowerPoint, (ii) arrange the elements spatially and add 3D models using the PowerSpace Editor based on the slide concept, (iii) define the order and relations between the slides using the PowerSpace Editor, and finally (iv) evaluate the authoring result using the PowerSpace Viewer.

DART is based on the Macromedia Director, and it has three main features as compared to PowerSpace. Firstly, informal content (e.g., sketches) can be added into the AR environment in addition to the polished content (e.g., CAD models). Secondly, synchronized capture and playback of tracking, video and other sensor data are supported so that AR designers can test the authoring with real setups early and often. Thirdly, DART provides an event-based concept to define the rendering of the authored contents as compared to the slide concept of PowerSpace. The event-based concept enables the AR designers to define various cues (e.g., a particular marker is detected in the view) that triggers the actions (e.g., playing an audio clip), and such concepts are more flexible for the rendering of the authored contents.

In spite of the benefits, utilizing existing authoring tools brings certain technical constraints. The slide concept of Microsoft PowerPoint limits the flexibility of the data structure of the authored contents, and the inadequate support of 3D models and multi-user applications of Macromedia Director limits the popularity of DART.

Certain authoring tools have been specifically designed for particular application domains by analyzing their unique features [Haringer and Regenbrecht 2002, Grimm *et al.* 2002, Knopfle *et al.* 2005, Güven and Feiner 2003]. Sharing the slide metaphor of PowerSpace, CATOMIR [Grimm *et al.* 2002] emphasizes the hierarchical structure of the instructions of assembly tasks, and it enables the users to compose AR-based instructions efficiently. A template-based authoring method has been proposed for the application domain of maintenance and repair [Knopfle *et al.* 2005], where all the maintenance tasks are built on atomic operations. These operations can be classified into different predefined categories, and specific operations within each category only differ in certain parameters. For example, *release fastener* is typical maintenance category, and its corresponding operations can differ in the kind of fasteners, e.g., screw, clamp and bolt. Each category of maintenance operations can be developed as a template in the reported system. Thus, the user can initialize templates, parameterize them, and define their temporal order. This approach fits well in the way that the technical writers work, i.e., considering the operations that need to be carried out to solve the maintenance problems and defining the values of the variables for each operation.

The MARS (Mobile Augmented Reality System) Authoring Tool [Güven and Feiner 2003] is another domain specific AR authoring tool for developing 3D hypermedia narratives, which can be used for city touring, etc. Using this tool, the user can position media files spatially using the 3D model of the application environment and temporally using a timeline, and create hypertextual links among the virtual contents. These domain-specific authoring tools are typically useful for their designated domains, but lack generality.

While these reported authoring tools provide 2D desktop GUIs for the users to develop AR applications, Lee and Kim [Lee and Kim 2009] proposed the concept of immersive authoring, which is similar to the concept of ―What You See Is What You Get (WYSIWYG). They argued that 2D desktop GUIs cannot provide direct and intuitive specifications of the spatial aspects of the authored contents, such that excessive trial-and-error may be needed. The key idea of immersive authoring is that the authored content should be checked first hand and *in situ*, i.e., the author is able to create and verify the content in the same environment as the one that the AR content is finally rendered. A prototype immersive authoring system, iaTAR, has been built. In iaTAR, physical props are used as a medium to select and control virtual objects. Their user study indicated that immersive authoring has significant advantages in specifying spatial arrangements, but unsuitable for abstract tasks, e.g., logical programming, as compared to 2D desktop GUIs. Lee *et al.* [Lee *et al.* 2011] improved the immersive tangible user interfaces by using RFID (Radio Frequency Identification) technology to link the virtual and real objects, and proposed the context-adaptive tracking method that solves the problem of markers being obstructed by the users during the authoring process. ARtalet [Ha *et al.* 2010] allows the users to manipulate the trajectories of 3D objects and provide audio/vibration feedback.

Online authoring in unknown environments is another aspect of AR authoring that is being addressed recently. All the above-mentioned authoring tools only support offline authoring, and cannot be used for real time generation of AR contents in collaborative work. Abeykoon *et al.* [Abeykoon *et al.* 2012] proposed a real time authoring tool for supervising groups of users in an AR environment. The system comprises of a CAD model of the application environment and uses natural features tracking to allow users to author virtual contents which can be visualized. An online authoring tool that removes the need to construct a CAD model of the application environment in advance has been reported, and it can be used to create AR-based instructions online in unknown environments to support remote collaborative work [Reitmayr *et al.* 2007].

The system reported by [Reitmayr *et al.* 2007] is able to estimate and measure the location and shape of the real world structures using simultaneous localization and mapping (SLAM), and such information can be used to register virtual contents in the real world. Using this system, a remote user can instruct a local operator to perform various tasks by placing annotations on the real world geometrical structures in the live video. For example, to instruct the operator to lift a cup on the desk, the remote user can place a circular shape on the cup in the current frame of the video stream to highlight it, and the system will measure the shape of the cup in the incoming video frames and highlight it using an ellipse. However, only simple annotations (e.g., circles and polygons) can be used in the system.

2.3.2 Summary

The reported AR authoring tools have made authoring easier for AR developers through various aspects. However, there are several research issues that need to be addressed.

1) A standard format for the authored contents is not yet available, and different formats have been used in different authoring tools. This issue hinders the file sharing and authoring collaboration between AR developers, and also poses challenges to the rendering engines in the AR systems.
2) Informal media files that can be created using traditional approaches, e.g., sketches, are generally unsupported in the reported AR authoring tools, and only the polished content, e.g., images and virtual models, can be added into the AR environment. However, some AR developers are used to forming and testing their ideas using the informal media files. In addition, supporting such files can attract more users, who have little technical knowledge to construct the polished contents to participate in the authoring.

3) Human-system interaction remains indirect and inefficient, as it is unintuitive to position and orientate virtual objects through 2D desktop user interfaces and difficult to conduct logical programming using 3D immersive interfaces.
4) Unknown environments pose challenges to AR authoring, and the tool reported in [17] is still preliminary. It can only estimate and measure simple geometric features, namely, circles and polygons, and only simple annotations can be added.
5) In the reported work, heavy emphasis has been placed on providing intuitive interactions for the users to add and arrange the virtual objects spatially in the AR environments, but the work on the logical programming aspect is still preliminary. However, logical reasoning is crucial for context-awareness. Thus, there is yet an authoring system which can be used for developing context-aware AR applications.

In this research, two of the research issues identified above will be addressed. Firstly, online authoring in unknown environments will be worked on so that AR-based visual interactions will be used in remote collaboration. Secondly, the logical programming aspect of AR authoring is explored, and a bi-directional tool for authoring context-aware AR contents is proposed.

Chapter 3 The ARAMS System

In this chapter, the features and architecture of the proposed AR-assisted maintenance system (ARAMS) are presented. In addition, the system software and hardware and the research tasks involved to achieve ARAMS are discussed.

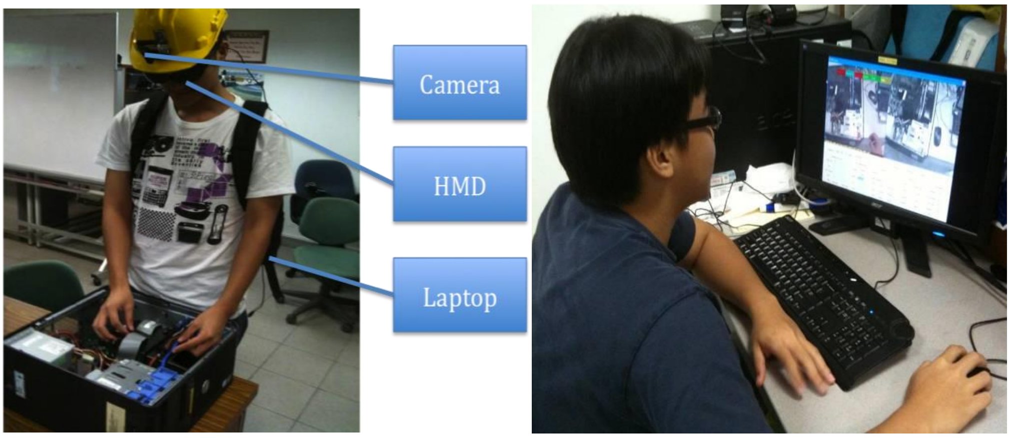
3.1 Introduction

ARAMS is to assist the maintenance technicians in routine and *ad hoc* maintenance activities by providing (1) context-aware information to the technicians, (2) a mobile user interface that allows the technicians to interact with the virtual information rendered, (3) a remote collaboration mechanism that allows the expert to create and provide AR-based visual instructions to the technicians efficiently and effectively, and (4) a bi-directional content creation tool that allows dynamic AR maintenance contents creation offline and on-site. ARAMS can analyze the contexts of maintenance tasks to provide relevant and useful information to technicians through registering and rendering the information on the real equipment correctly. Thus, the system can enhance the efficiency and productivity of maintenance operations. In addition, and a content development tool is provided to allow engineers and technicians to construct context-aware AR-based maintenance contents on-site or offline that can be adapted to the various contexts.

3.2 Software and Hardware in the System

The ARAMS system is developed using C++ and Java in Visual Studio 2005 and Eclipse, and the programs are integrated through socket communications. The system employs s few open source APIs (Application Programming Interfaces) and libraries, e.g., the OWL API [OWL API] for ontology construction and logical rules definition, the Pellet reasoner [Pellet] for context management, OpenGL for rendering the virtual objects in the real environment, and OpenCV [OpenCV] for some CV functions.

The ARAMS system provides a mobile platform (Figure 3.1(a)) for the maintenance technicians, and the platform consists of a camera, an HMD, and a laptop. The camera captures live video of the maintenance scenes, the HMD displays the AR environment to the technician, and the laptop runs the ARAMS system. In addition, a desktop equipped with a mouse and a keyboard is provided for the remote expert (Figure 3.1(b)).



(a) Mobile platform (b) Desktop-based system

Figure 3.1: System hardware

3.3 System Architecture and Research Tasks

The ARAMS system (Figure 3.2) consists of

(1) On-site authoring (OSA) for maintenance technicians to create, edit and update AR contents;

(2) Offline authoring (OFA) for maintenance experts to develop context-aware AR maintenance contents; OFA and OSA form the bi-directional tool;

(3) Online authoring (ONA) for experts to create AR-based instructions during remote maintenance activities;

(4) Database stores virtual and AR maintenance contents;

(5) Context management (CM) collects and reasons maintenance contexts;

(6) Tracking and registration;

(7) AR-based visualization (ARV) for tracking and rendering the AR contents in the maintenance environments.

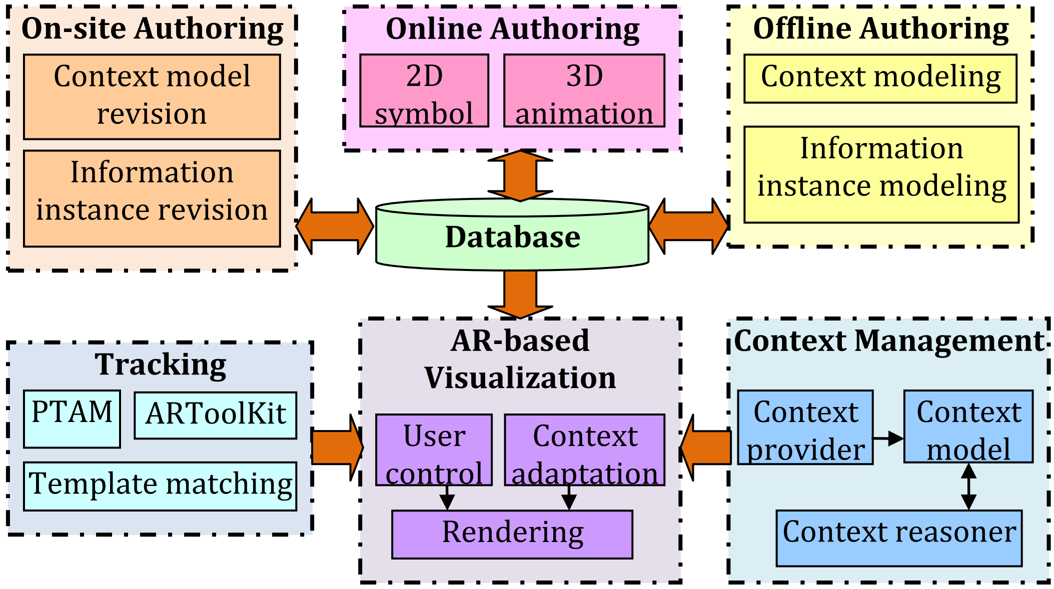


Figure 3.2: System Architecture

3.3.1 Online authoring

ONA is designed to allow remote expert engineers to develop and augment AR-based maintenance instructions to assist technicians in remote maintenance.

To develop an effective authoring system for remote maintenance, several issues need to be addressed. Firstly, the AR-based instructions should be generated online easily, i.e., online authoring. Remote maintenance is frequently an adjust-observe-readjust loop until certain parameters have been achieved, e.g., adjusting valves to obtain the correct pressure. Thus, it is important to enable the experienced operator to create instructions according to different conditions online instead of preparing them in advance. For example, the experienced operator can highlight a button by annotating this and ask the technician to press it.

Secondly, the requirement of preparing the maintenance environment in advance, i.e., attaching markers at specific locations or constructing virtual models of the environment for tracking and registration, should be removed. Preparing the maintenance environment can be time-consuming and costly, and various policies, such as safety regulations, may preclude modifying the environments. Thus, this issue hinders the mobility and usability of the AR-assisted system.

Finally, a stable authoring interface should be provided as authoring on the live video can be unstable and error prone. In the reported systems [ARTESAS, Harmo *et al.* 2001, Reitmayr *et al.* 2007], the live video transmitted from the maintenance environment remains the main channel for the expert to access the environment and perform authoring. The camera capturing the live video is usually worn on the technician, e.g., being part of the HMD, and movements of the camera are unavoidable which disrupt the authoring process of the expert. To allow the experts to access the remote environment easily through live video and also provide them with a stable authoring interface, a novel interface compromising both live video and static authoring interface (Chapter 4) is designed.

3.3.2 Bi-directional context-aware AR contents authoring

As there is a lack of a user-friendly authoring tool that can be used to create context-aware AR contents easily, a bi-directional maintenance content development tool is designed and developed for AR developers and maintenance technicians to develop context-aware AR contents in this research.

In this research, the concepts and techniques to author AR contents for context-aware AR applications are proposed. In addition, a bi-directional authoring tool comprising the OFA and OSA modules of the ARAMS system is developed. In the proposed tool, a 2D GUI will be provided to abstract the low-level programming techniques so that the users, who are not programmers, can participate in the authoring process easily. In addition, an intuitive mobile user interface will be developed for the maintenance technicians to interact with and author AR contents on-site. As the mobile user interface shall allow the technicians to conduct both 2D (e.g., inputting texts) and 3D tasks (e.g., arranging virtual models spatially) with ease, and 2D (e.g., mouse and keyboard) and 3D interaction tools (e.g., 3D mouse) are suitable for 2D and 3D tasks respectively [Lee and Kim 2009]. An interaction tool that can be used for both 2D and 3D interactions is designed in the mobile user interface (Chapter 4).

3.3.3 Context management

The CM module is designed to collect low-level contexts, e.g., the user's location, from the user inputs and sensors in the environment, derive the high-level contexts, and process the context knowledge queries from the ARV module. It consists of the context provider, context model and context reasoner.

Ontology is used to model the contexts and define the intrinsic semantics of the maintenance environment; instances of concepts (class, subclass, and property) can be edited to construct an ontology for a specific maintenance application.

The context provider acquires and interprets the context data and information from the user inputs and various other data sources, e.g., web servers, digital timer, and physical sensors. After acquiring and interpreting low-level contexts from the context data and information, the context provider will encode them in OWL (Web Ontology Language) descriptions. The acquired low-level (direct) contexts will be transmitted to the context reasoner to infer the high-level (deduced) contexts.

The context reasoner infers and derives high-level contexts from the low-level contexts and handles context knowledge queries from the ARV module.

3.3.4 Tracking

Tracking estimates the camera poses in real time so that the AR maintenance contents can be augmented to the technicians correctly in the real environment. Three tracking approaches are used, namely, ARToolKit for tracking predefined fiducial markers and registering the contents created using the bi-directional tool, PTAM for constructing 3D maps online and registering the 3D animations via the ONA module, and template matching [OpenCV] for searching the rectangular region in each frame of the live video and registering the 2D symbols on the matched region. ARToolkit has stable tracking performance, but it requires the markers to be visible in the camera view all the time. This is inconvenient in remote maintenance, where the technicians usually have to move about according to the experts' instructions. Thus, PTAM that requires no markers is employed in the ONA module.

3.3.5 Context adaptation and AR-based visualization

The ARV module employs context-awareness to improve the usability the AR system through adapting the virtual information provided to the various contexts so that it can be more useful to the users. For example, suitable level-of-detailed maintenance instructions should be provided to the maintenance personnel according to their level of expertise.

To achieve this, the ARV module consisting of user control, context adaptation and rendering is designed. Context adaptation manages the rendering of the context-aware AR maintenance information based on the real time contexts. The information relevant to the current contexts will be augmented in the real environment.

3.4 Summary

This research presents a novel real-time equipment maintenance system to assist the maintenance technicians in routine and *ad hoc* maintenance activities. The ARAMS system can provide useful information to the technicians according to their contexts, allow the technicians to interact with and edit the provided information, and support effective visual instructions-based remote collaboration. To achieve the proposed features, a system architecture is presented and the software and hardware are described. The research tasks involved include (i) developing an online communication interface for remote maintenance between equipment experts/vendors and maintenance personnel, (ii) developing a bi-directional authoring tool for creating context-aware AR maintenance knowledge, and (iii) developing the ARV module for adapting and rendering the authored context-aware AR contents to the maintenance technicians. These research tasks will be described in detail in chapters 4 and 5.

Chapter 4 Online Authoring for AR-based Remote Maintenance

In this chapter, the enabling techniques to achieve effective AR-based remote collaboration using the ARAMS system will be described. The real world scenario will be analyzed first, and the proposed concepts and methodologies to achieve an online communication interface for remote maintenance between equipment experts/vendors and maintenance personnel will be presented.

4.1 Real World Scenario Analysis

An analysis of the real world scenario is necessary for designing useful engineering tools. The application scenario of the proposed system is remote maintenance, where a remote expert instructs a technician to perform maintenance tasks. Before the emergence of the AR technology, speech-based communication dominates this collaborative work. However, certain operations are difficult to be described verbally and verbal instructions are not intuitive. In addition, speech-based communication should be kept to a minimum in a noisy environment. AR technology provides a promising instructional method for the tele-assistance as it enables the expert to provide visual instructions through an AR environment.

In maintenance, the technician could be someone who possesses the necessary basic knowledge of equipment maintenance or a novice. Besides, the maintenance tasks can range from simple operations (e.g., tightening a valve), which can be carried out by layman, to complex tasks (e.g., disassembling an equipment), which may require step-by-step detailed instructions from the expert.

When the operations are simple or the technician is experienced, speech-based tele-assistance can be effective. However, AR-based communication is an important alternative in this scenario. For example, as modern equipment usually comprises numerous components, highlighting certain components by augmenting 2D symbols on them is convenient and efficient for both the expert and the technician.

When the operations are complicated or the technician is untrained, detailed and intuitive instructions should be provided to decrease the cognitive load of the technicians and avoid any mistakes. The AR technology provides a solution by augmenting 3D animations of the virtual operations on top of the real parts to be worked on. Thus, the remote worker could view highlighted virtual models of the real components being operated and simply follow the operations [BMW Augmented Reality].

AR technology can be very useful in remote maintenance provided a user-friendly and efficient AR authoring system is available. Thus, an online authoring tool is proposed to enable the expert to develop AR-based manuals (2D symbols and 3D animations) and provide intuitive instructions to the maintenance technicians.

4.2 Dual Window User Interface

To extend the expert's view and improve the robustness of the authoring system, the Dual Window User Interface (DWUI) is proposed. As shown in Figure 4.1, DWUI is composed of three sections, namely, the left window displays the live video transmitted from the remote maintenance environment, the right window displays the keyframes built in the 3D map [Klein and Murray 2007], and the bottom window displays the menu for authoring. Live video captured by the remote worker's camera is the main channel for the experienced operator to obtain real time information of the remote worksite. Keyframes, which are captured at the different perspectives of the worksite [Klein and Murray 2007], provide a convenient way for the experienced operator to obtain a wider view of the operating environment. In Figure 4.1, the expert retrieves a keyframe providing a larger view of the printer while the other view shows the technician carries out the maintenance operations. Thus, this method relieves the technician from having to shift the camera to display the necessary information to the experienced operator.

Besides, the DWUI features more stable and accurate authoring as the authoring process is performed on still keyframes instead of the traditional live video. The expert can retrieve keyframes, which feature the best perspective for online modeling [Hengel *et al.* 2009], or an image template outlining, to create the 3D animation or 2D symbol augmentations. The rendering of the authored contents can be controlled via the authoring menu.

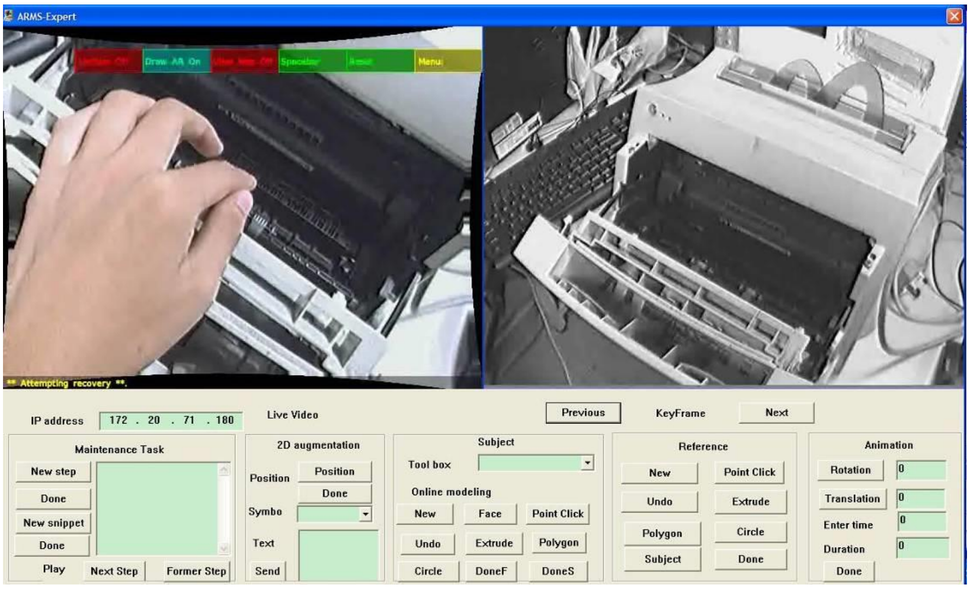


Figure 4.1: DWUI

4.3 AR-based Remote Maintenance Overview

The proposed AR-based remote maintenance scheme utilizes four modules within the ARAMS system, namely, the ONA module, the tracking module, database, and the ARV module (Figure 4.2). The tracking and ARV modules are at the technician’s site, while the ONA and database modules are at the expert’s site. The two sites are connected through wireless communications.

The tracking module provides the necessary information of the remote maintenance environment to the expert. The tracking module includes the 3D map and live video. The 3D map is created by adopting the tracking and mapping system [Klein and Murray 2007], and it comprises a bundle of 3D feature points and keyframes. Live video is captured using the camera that is mounted on the technician’s HMD to provide real time information of the maintenance environment.

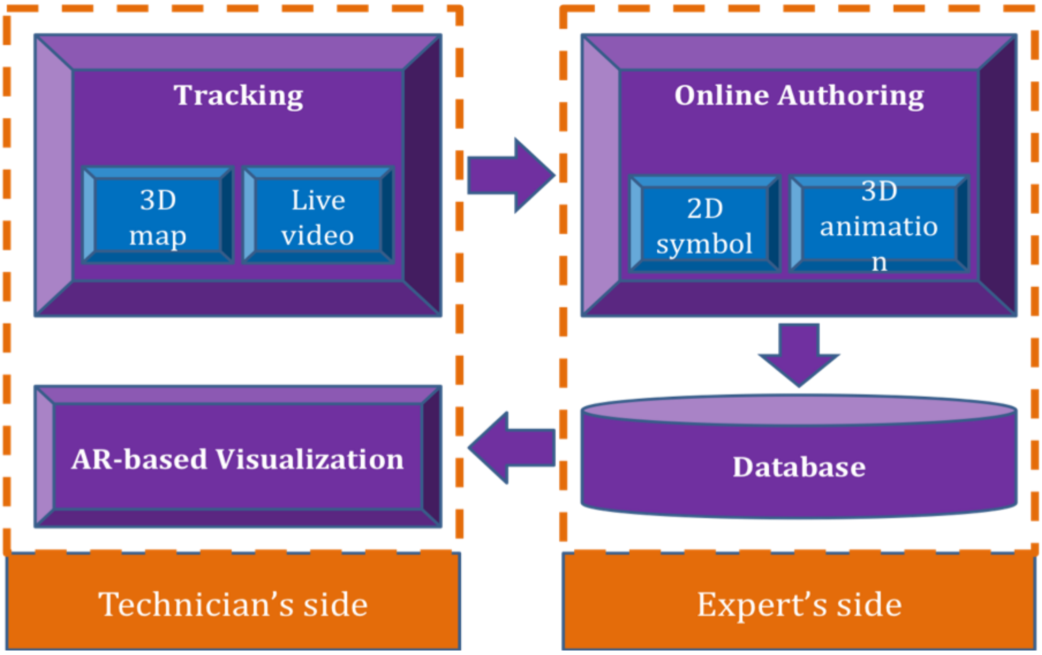


Figure 4.2: AR-based remote maintenance

The ONA module controls the process of creating AR-based instructions, 2D symbols and 3D animations. 2D symbols can be either selected from the pre-defined (e.g., rectangles) list or input by the expert (e.g., texts), and their registration positions in the image coordinates can be defined through outlining an image template in the keyframe. To create 3D animations, the expert needs to identify the 3D models and compile animations using the animation module. Virtual models of standard components (e.g., bolts and nuts) can be selected from the virtual model library, and the components, which virtual models are not available, can be modeled online. The animation module enables the expert to animate the operation process through defining certain parameters.

The database stores the authored AR-based instructions, i.e., 2D symbols and 3D animations. The ARV module renders the instructions for the technician.

4.4 Tracking

The tracking module captures and processes the live video of the maintenance environment. This module employs the Parallel Tracking and Mapping (PTAM) system [Klein and Murray 2007] to (i) track the real environment without the need of *a priori* knowledge, and (ii) process the live video to generate 3D feature points and keyframes for authoring. As shown in Figure 4.3, a 3D map comprising a bundle of 3D point features; keyframes can be created online and the camera pose is estimated by matching point features in the current frame with those in the 3D map. The keyframes are displayed in the right window of DWUI, where the expert can access the remote worksite and perform authoring. The 3D feature points will be used for online modeling [Hengel *et al.* 2009] besides tracking. Tracking, which accuracy depends highly on the quality of the video frames, is conducted on the technician's side, so that the video frames to be transmitted to the expert side can be compressed at a relatively higher rate. This approach reduces the amount of information to be transmitted and decreases the transmission time between the two sides, i.e., video transmission delays.

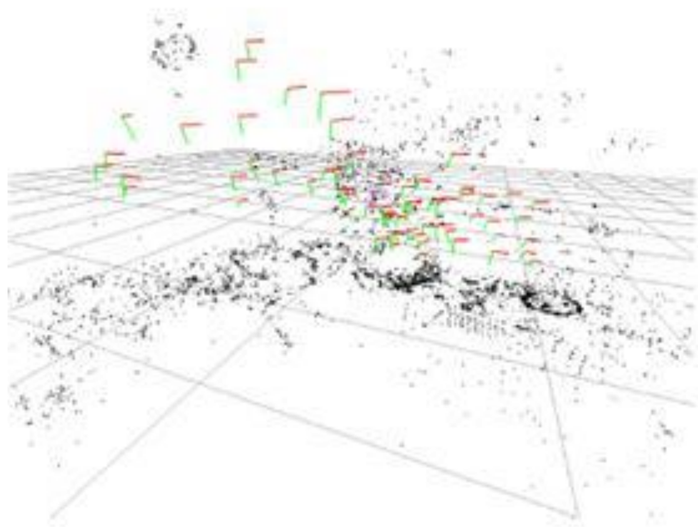


Figure 4.3: 3D map comprising keyframes and feature points [Klein and Murray 2007]

4.5 Online Authoring

The ONA module is the core of the AR-based remote maintenance, and it manages the authoring of 2D symbols and 3D animations.

4.5.1 Data structure

As the application of the proposed system is remote maintenance, the data structure is designed according to the configurations of the maintenance tasks. In typical equipment services, maintenance tasks usually consist of a number of steps, and each step is composed of certain operations. Thus, *Task*, *Step*, *Operation*, and *Timeline* are introduced to define the data structure of maintenance operations and tasks.

Task is composed of a series of *Step*s, and the transition between *Step*s is controlled online, i.e., the animation of a particular *Step* is looped over and over until the expert issues a command such as “next”. Thus, the expert can control the maintenance procedures by developing and augmenting certain virtual animations. The augmentation is looped over and over for the technician to understand the conveyed instructions clearly.

Each *Step* consists of one or a few specific *Operation*s, and the *Timeline* of each *Step* consists of the temporal relationship between the *Operation*s (Figure 4.4). The *Operation* is the basic element of the data structure, and it represents the manipulation of one or a few components. An *Operation* can either be the augmentation of a 2D symbol or a 3D animation. Outlined image templates and 2D symbols for 2D symbol augmentation, and virtual models and animation related parameters for 3D animation augmentation are stored in each *Operation*.

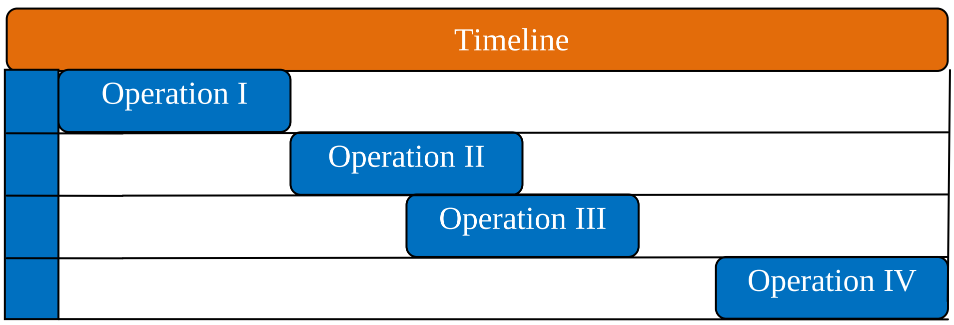


Figure 4.4: Configuration of *Step*

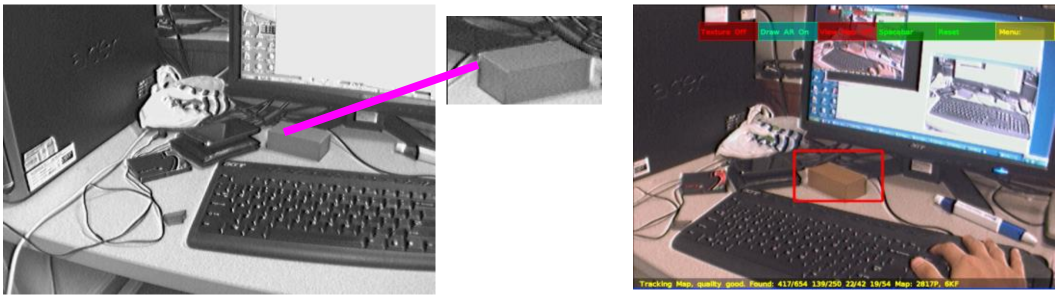
4.5.2 2D symbols

This section presents the process of creating 2D symbol augmentation on the live video. The process to define a 2D symbol augmentation includes two steps: (i) choose (or input) the symbol and (ii) identify the 2D position in the image view where the symbol will be augmented.

To identify the 2D position for augmentation, the expert can outline a rectangular region in a keyframe (Figure 4.5(a)). The selected region is stored as a rectangular image template and searched in each frame of the live video using the OpenCV [OpenCV] function cvMatchTemplate. The template matching algorithm establishes the correspondences between primitives, either as grey images or features, extracted from a template and digital images [Heipke 1996]. During the matching process, a template window moves within a live video frame, and the cross correlation coefficient is calculated between them. The best matching position is where the maximum coefficient is estimated.

The matched region in the current frame will be the position where the symbol is to be registered. As the 2D tracking is based on image template matching, to improve the stability and robustness of the augmentation, the expert should select the keyframe that is closest to the current frame (in terms of camera position) to outline the rectangular region.

The symbols can be either selected from a pre-defined list in the system (e.g., a rectangle) or input by the expert (e.g., input a series of characters). In Figure 4.5, the selected template (a) is searched in the current frame, and a red rectangle is augmented on the matched region.



(a) (b)

Figure 4.5: (a) Box selected in the keyframe, and (b) Red rectangle augmented in the current frame to outline the selected box

4.5.3 3D animations

This section presents the process of creating 3D virtual animations of the maintenance operations. The process to create a 3D animation includes two steps, namely, (i) identify the virtual models (online modeling or choose from the virtual model library), and (ii) define the animation parameters.

4.5.3.1 Virtual model selection

Virtual models are indispensable for creating 3D animations, and the database provides CAD models of standard parts (e.g., bolts and nuts) and common mechanical tools (e.g., screwdrivers and wrenches). These CAD models are saved as STL files so that they can be drawn in the AR environments using OpenGL. The expert can retrieve these pre-constructed models to create 3D animations.

However, the CAD models of most components are usually unavailable. In addition, it is inconvenient and may be costly to pre-construct them. Recent development of the CV technology has provided methods for online modeling objects based on video frames. In [Hengel *et al.* 2009], an *in situ* image-based modeling system is proposed. It enables users to create accurate 3D models of real objects in the scene efficiently. In addition, it allows "on demand" modeling, i.e., creating minimal 3D structures needed for a particular application. Subsequently, the user is able to interact with the real objects intuitively. As shown in Figure 4.6, the CD tray of a desktop can be modeled on the still images.

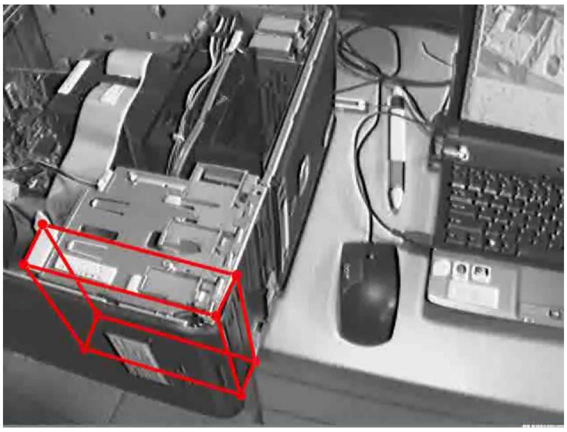


Figure 4.6: Part of the modeling process of a CD tray

For remote maintenance, the expert may not need to model all the components completely and precisely as long as they can be recognized and understood by the technician. In addition, the created virtual models should be of the same size as their real counterparts. The modeling process is carried out on the keyframes displayed in the right window of DWUI, and the created 3D models are stored for animation creation.

4.5.3.2 Animation parameters

After identifying the virtual models (choosing from the virtual model library or online modeling), the expert will proceed to animate the operations of the components. In equipment maintenance, rotation and translation are the two main motions of the components. Thus, the prototype system supports the animation of these two motions. Two templates are employed in the system, and the expert can create animation by parameterizing the templates.

A few parameters are introduced to define an animation, namely, *Enter time*, *Exit time*, *Distance* (translation), *Degree* (rotation), *Original position*, and *References*. *Enter time* is the time the animation is augmented in the scene (Figure 4.4), and *Exit time* is the time the animation is removed from the scene (Figure 4.4). The range between *Enter time* and *Exit time* is the time range of the animation in the *Timeline*. *Distance* (translation) and *Degree* (rotation) define the range of the motion carried out on the virtual model between the *Enter time* and *Exit time*.

Original position is the 3D position of the virtual model at the beginning of the animation. If this virtual model is modeled online, the 3D position of the real part will be defined as the *Original position* automatically. If the models are chosen from the virtual model library, the expert can locate the *Original position* easily by matching the features between the virtual and real objects. For example, the expert can position a virtual screwdriver along the axis of a real bolt by matching their centerlines. The features of the real objects can be identified using the online modeling system [Hengel *et al.* 2009]. *Reference* is the motion direction of the virtual object, i.e., translation and rotation directions. As the components are usually interdependent (e.g., a nut rotates about the axis of a bolt), *Reference*s are defined based on the real components (e.g., identify the *Reference* as the rotation direction of a virtual nut by locating the axis of a real bolt [Hengel *et al.* 2009]) in this prototype system.

4.6 Database and AR-based Visualization

An *Operation* is represented by a particular animation, and a *Step* is composed of a certain number of *Operation*s (Figure 4.4). Created *Step*s are stored accordingly in the database. The ARV module renders the authored contents to the technician using OpenGL. The rendering is controlled by the expert, and he selects the *Step* to render. A *Step* will be looped over and over until the user issues the command "Next Step". Besides, the previous *Step*s can be retrieved to check the maintenance services via the command "Former Step".

4.7 Summary

This chapter presents the proposed methodologies for enabling effective AR-based remote maintenance. With the integration of robust tracking and efficient image-based online modeling, an intuitive user interface DWUI has been developed and used to construct the prototype system, which enables the expert to create AR-based instructions conveniently and provide intuitive instructions to the technician. The ARAMS system (i) provides convenient and efficient methods to create intuitive instructions, i.e., 2D symbols and 3D animations online, (ii) requires no *a priori* knowledge of the environment, (iii) improves the robustness of the authoring process by using still keyframes based authoring, and (iv) enables the expert to access the remote operating environment conveniently via the keyframes of the live video.

Chapter 5 Context-aware AR Contents Authoring and Visualization

In this chapter, the proposed concepts and methodologies of authoring and adapting context-aware AR contents will be presented. The real world scenario will be analyzed first, and the proposed bi-directional authoring tool for developing context-aware AR information and the ARV module for adapting and rendering the contents to the technicians will be described.

5.1 Real World Scenario Analysis

Using AR, the equipment maintenance information, which is recorded in paper-based or electronic manuals traditionally, can be registered on the real equipment virtually. However, AR maintenance systems should be more than just an advanced data presentation engine as compared to the paper-based and electronic manuals; they should evolve from passive manuals to active information providers, namely, filtering the useful information and rendering the information properly to the technician by analyzing the contexts.

To enable AR maintenance systems to be active information providers, the maintenance information in these systems that are needed in the various maintenance activities should be analyzed. Generally, the maintenance information required can be classified into the recommended conditions on which the maintenance activities should be conducted (When) and the instructions on carrying out these activities (How). These conditions and instructions vary according to the types of the maintenance activities. For preventive maintenance where activities are carried out to prevent faults from occurring, a list of recommendations is usually provided by the manufacturers to create the activities for the individual components separately. These recommendations generally comprise of pairs of parameters and conditions. The parameters can be equipment operating hours, sensor readings, vehicle distance traveled, etc., and the conditions can be frequency (e.g., operation periods), threshold (pressure higher than certain values), etc. When the conditions are met, the corresponding maintenance activities should be conducted. The instructions on the preventive maintenance activities are usually provided, and they are sequences of several steps.

For corrective maintenance where activities are carried out to restore a piece of equipment that has broken down from an operational condition, a list of equipment failures is usually provided. In cases of breakdown, the activities corresponding to the failures should be conducted. The instructions provided on the corrective maintenance activities are usually composed of possible causes of the failures and the corresponding actions to be taken.

Thus, an active maintenance information provider should provide alerts on the maintenance activities and useful instructions according to the contexts, instead of requiring the technicians to check whether certain maintenance activities should be carried out and look up instructions from the manuals manually. To achieve this, the contexts of the maintenance technicians need to modeled and analyzed, so that context-relevant information can be provided to assist the technicians.

5.2 Context-aware AR Contents

AR has been used as an interface to register virtual objects in the real environment, and these virtual objects can be used to describe certain information to the users. Generally, different combinations of virtual objects can be used to convey the same information and providing information with the suitable content and presentation to the users is the aim of integrating context-awareness with AR.

5.2.1 Information instance

In ARAMS, an information instance is a combination of virtual objects which can be authored and rendered independently. The system consists of many information instances. Providing context-aware AR services to a user involves selecting appropriate combinations of virtual objects (information instances) and adapting the display of the selected information instances based on the user's contexts.

An information instance has the following context-adaptable properties that are used to relate the content and rendering of the information instances to the contexts:

1. Content-related properties: these properties characterize the information that is described in the information instance and the group of virtual objects used to describe this instance; they allow the content of an information instance to be linked to the contexts.
2. Rendering-related properties: these properties characterize the rendering of the virtual objects, and relate the presentation of the information instances with respect to the contexts.

The content-related properties are item and presentation. Property item represents an instance of the domain knowledge, e.g., instructions on the lubrication of a piece of equipment, and it can be related to the contexts of the various statuses of a task, the user's interest, *etc.* The presentation property refers to the combinations of virtual objects used to represent the item, and it can be intuitive presentation, symbolic presentation, and/or textual presentation. Intuitive presentation uses graphics and 3D models to describe the information intuitively, symbolic presentation uses symbols, e.g., arrows, and textual presentation uses texts. The selection of the presentation type should take into consideration the user's profile, the capability of the mobile device, tracking accuracy, etc. For example, intuitive presentation is more appropriate for a novice as it is easier to be comprehended, textual presentation which consumes less memory than rendering complex 3D models should be used when limited resources are available from the mobile device, and 3D object models can be replaced with callout texts and lines when the tracking accuracy is low [Coelho *et al.* 2004].

Rendering-related properties refer to the display of the information instance in the real environment, and they are classified into two categories, namely, Format and Registration. Format manages the visual properties of the augmented information, and it consists of color, transparency, etc. Based on the requirements of an application, certain properties can be defined and adapted to the contexts in this application, *e.g.*, user's preference.

Registration influences the registration of the virtual content in the AR environment, and this category of properties include translation, rotation, and scale. These properties can also be adapted to specific contexts, e.g., the scale of the virtual objects can be adapted based on the distance of the objects from the user so that the objects can appear larger when the user moves closer to the objects.

5.2.2 Adaptation scheme

The adaptation scheme to provide context-aware AR contents is illustrated in Figure 5.1. It is a two-step process, i.e., content adaptation and rendering adaptation. Content adaptation selects appropriate combinations of virtual objects (information instances) according to the contexts of the user, and rendering adaptation adjusts the rendering-related properties of the information instances according to the user's contexts. The adaptation process is conducted using the adaptation engine and is based on the logical rules that have been defined and the real-time information of the user's contexts. There are two sets of logical rules, namely, content adaptation rules which associate the content-related properties with the relevant contexts and the rendering adaptation rules which adjust the rendering parameters according to the contexts.

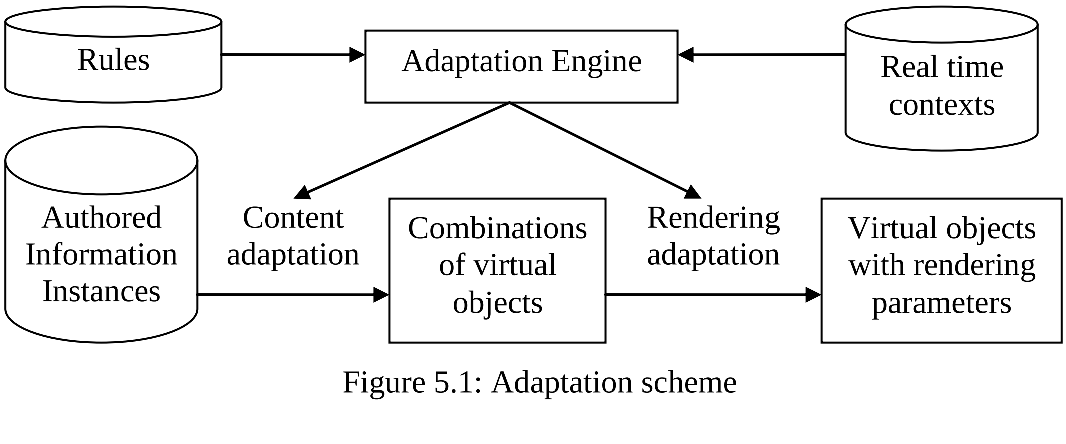


Figure 5.1: Adaptation scheme

5.3 Authoring and Visualization Overview

The ARAMS system provides functional modules for the users to author essential information for developing context-aware AR applications (Figure 5.2). These modules perform modeling of the contexts of the application environment, creating information instances, and adapting the authored information instances in order to provide appropriate AR contents that are relevant to the users' contexts.

As shown in Figure 5.2, five modules of the ARAMS system are used for context-aware AR information authoring and visualization, namely, the CM module, the ARV module, Database, the OFA module, and the OSA module. The OFA module provides a desktop 2D user interface for the AR developers to create context-aware AR contents through modeling the contexts and authoring the information instances, and the OSA module provides a mobile user interface for the maintenance technicians to perform authoring on-site. The OFA and OSA modules form the bi-directional authoring tool for developing context-aware AR information. The Database stores the context model, information instances, as well as the media files, e.g., images and CAD models. The CM module collects and interprets the raw data from various sources to form low-level contexts, derives the high-level contexts, and transmits all the contexts to the ARV module. The ARV module filters and retrieves the information instances that are relevant to the contexts, adjusts their rendering parameters, and displays the virtual objects in the AR environments.

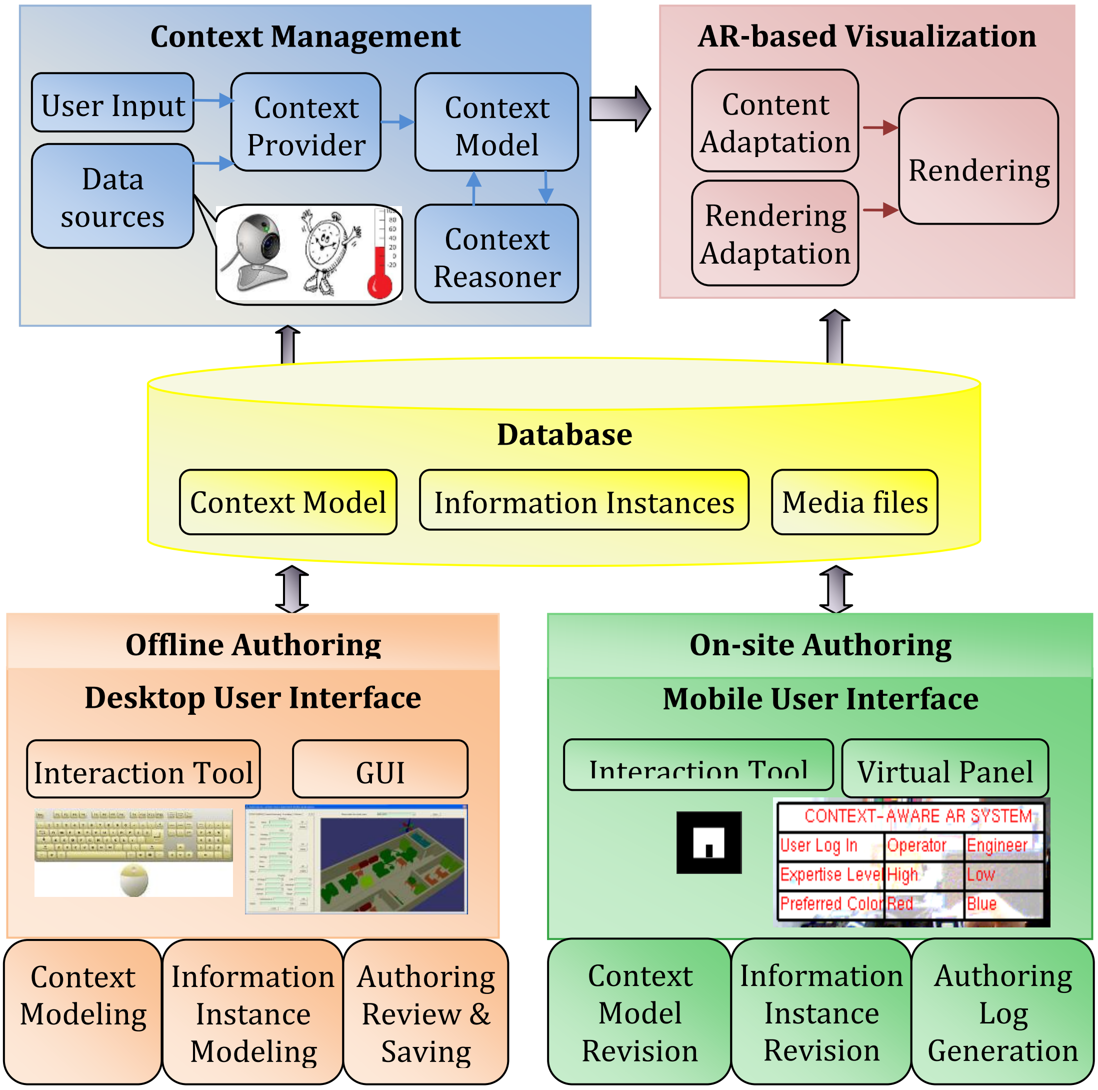


Figure 5.2: Context-aware AR contents authoring and visualization

5.4 Bi-directional Authoring Tool

The bi-directional process (Figure 5.3) consists of two main steps, namely, context modeling and information instance modeling.

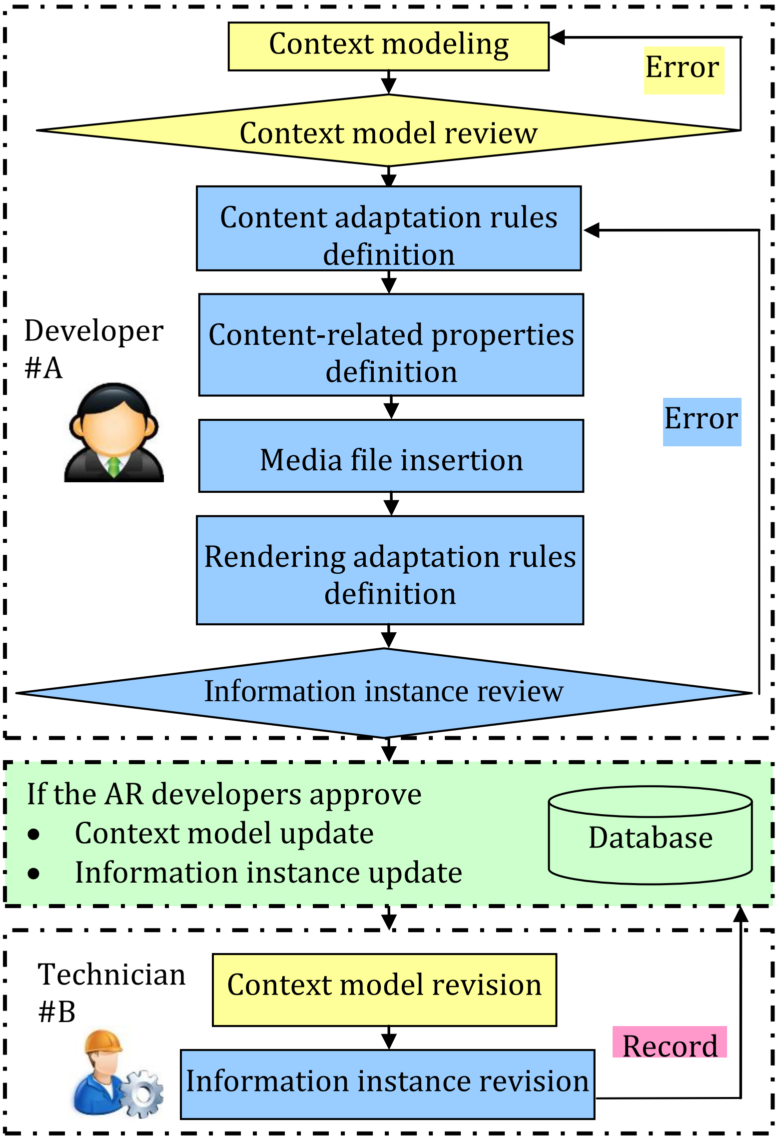


Figure 5.3: Bi-directional authoring

Using the OFA module, the AR developers can model the contexts of the application environment by constructing ontology and defining logical rules. The information instances of various maintenance procedures and information can be created by inserting various media files and relating their properties to the modeled contexts. In addition, a review function is provided for the developers to check the context adaptation of the created information instances, i.e., whether the information instances can be filtered and rendered according to the contexts.

Maintenance technicians can edit the context model and information instances via the mobile user interface in ARAMS. A data creation log recording details of the content development will be generated and sent to the AR developers for confirmation. If the content development is approved, the context model and information instances will be updated in the maintenance database.

5.5 Offline Authoring

5.5.1 User Interface

A 2D user interface consisting of two parts, namely, the authoring panel on the left and the virtual scene on the right (Figure 5.4) is provided for offline authoring. This panel provides command controls for the user to construct the ontology, define logical rules, and create information instances. The virtual scene displays the 3D model of the application environment, and the user can use the mouse to interact with the virtual scene, e.g., translate, rotate, and select the virtual models.

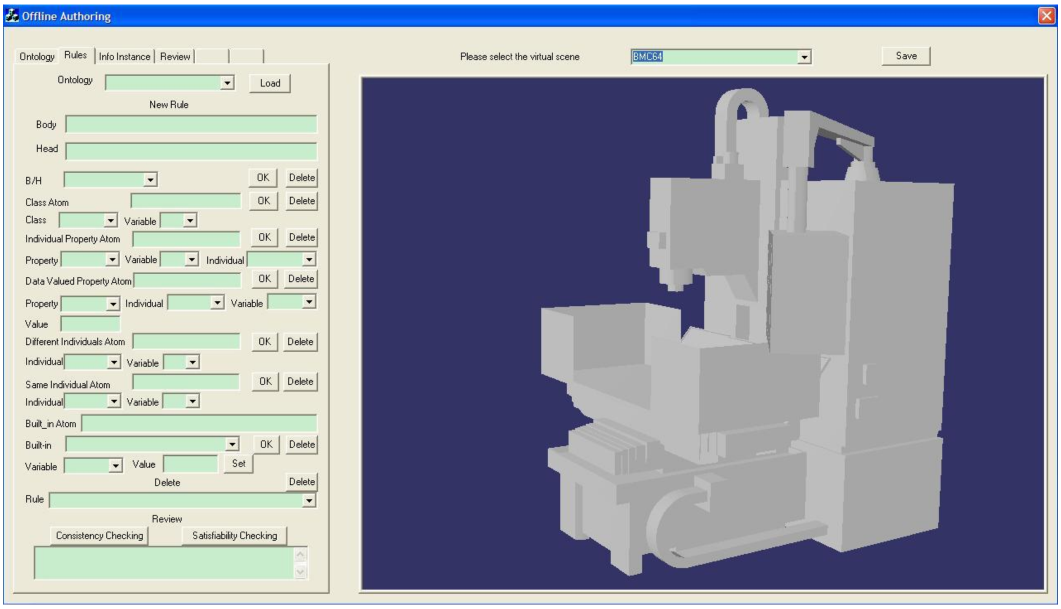


Figure 5.4: User Interface

5.5.2 Context knowledge authoring

Context modeling is an essential step for developing context-aware applications, and it is also the first step of the authoring process. Using the ARAMS system, the contexts can be modeled by constructing an ontology and defining logical rules.

5.5.2.1 Ontology construction

Ontology has been widely used in context modeling as it is independent of programming languages, enabling context reasoning using first-order logic, etc. A Context Ontology for AR Environments (COARE) (Figure 5.5) is derived from CONON [Wang *et al.* 2004] to model the contexts. CONON is a collection of OWL encoded RDF (Resource Description Framework) triples, and supports logic-based context reasoning. There are three types of concepts in COARE, namely, class, subclass and property. A class can have a number of subclasses, and each class/subclass can have several properties.

COARE consists of five classes, namely, CompEntity, Location, Activity, Person and InformationInstances. In CompEntity, the physical sensors and mobile devices in the environment can be registered and modeled. Property translation and rotation encode the 3D coordinates (position and orientation) of the camera, marker, etc., and they can be used to register virtual objects in the real environment. Location has the properties of Name, Translation, Time, etc. Location has two subclasses, namely, OutdoorSpace and IndoorSpace, and different locations can be registered so that location-aware services can be provided. For example, through identifying whether the user is located in IndoorSpace or OutdoorSpace, different tracking approaches can be used for AR implementation, e.g., GPS can be used in outdoor spaces but not in indoor spaces. The Activity class registers the tasks in the applications, and the Person class has the properties of ExpertiseLevel, PreferedColor, etc.

The InformationInstances class is used for modeling the AR contents, and all the AR contents (information instances) that have been authored will be registered in the ontology. This class has a display property in addition to the properties discussed in the Section 5.3, e.g., item, presentation, color, scale, etc. The display property is a Boolean value indicating whether the information instance should be rendered in the contexts.

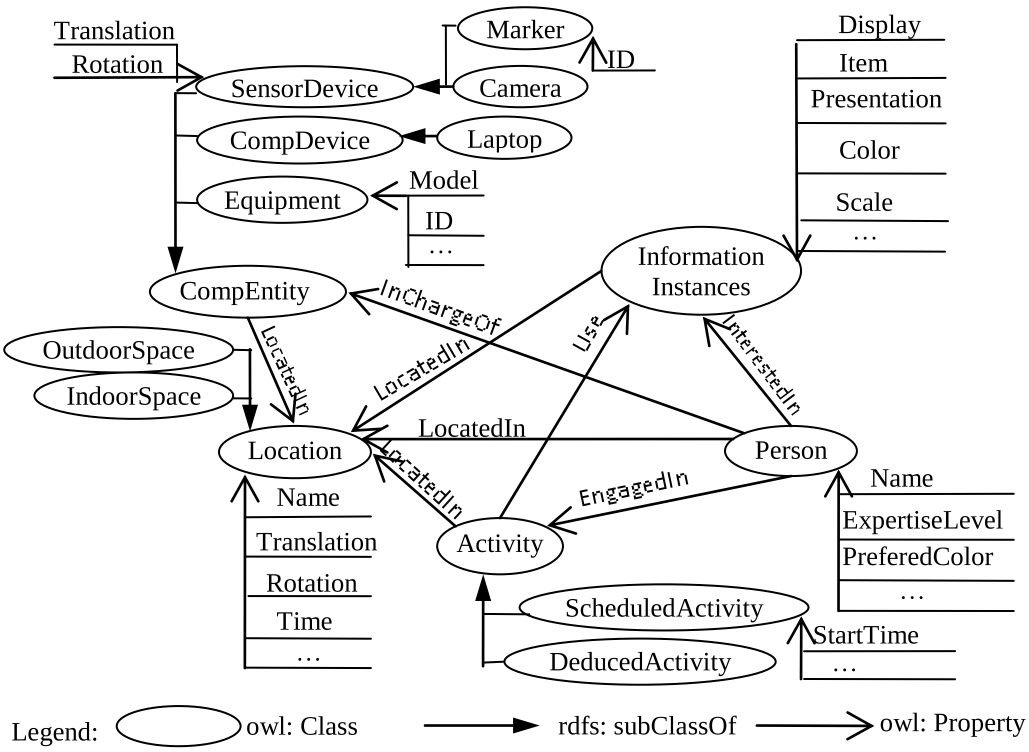


Figure 5.5: COARE

Thus, COARE provides a basis for context management in AR environments. To construct an ontology for a specific application, the instances of the concepts (class, subclass, and property) can be inserted or deleted to extend the upper ontology. For example, the maintenance task CheckOilLevel can be added as a subclass of Class ScheduledActivity with the properties of Status and StartTime (Figure 5.6(a)), the surrounding of the cabinet is inserted in the class of IndoorSpace (Figure 5.6(b)), and the marker Hiro is registered with the property of ID and the Boolean valued property InView indicating whether the marker is in the user’s view (Figure 5.6(c)).

```
<!-- COARE.owl#CheckOilLevel -->
<owl:NamedIndividual rdf:about="COARE.owl#CheckOilLevel">
  <rdf:type rdf:resource="COARE.owl#ScheduledActivity"/>
  <rdf:type>
    <owl:Restriction>
      <owl:onProperty rdf:resource="COARE.owl#Status"/>
      <owl:allValuesFrom rdf:resource="http://www.w3.org/2001/
      XMLSchema#string"/>
    </owl:Restriction>
  </rdf:type>
  <rdf:type>
    <owl:Restriction>
      <owl:onProperty rdf:resource="COARE.owl#StartTime"/>
      <owl:allValuesFrom rdf:resource="http://www.w3.org/2001/
      XMLSchema#dateTime"/>
    </owl:Restriction>
  </rdf:type>
</owl:NamedIndividual>
```

(a)

```
<!-- COARE.owl#Cabinet -->
<owl:NamedIndividual rdf:about="COARE.owl#Cabinet">
  <rdf:type rdf:resource=" COARE.owl#IndoorSpace"/>
</owl:NamedIndividual>
```

(b)

```
<!-- COARE.owl#Hiro -->
<owl:NamedIndividual rdf:about=" COARE.owl#Hiro">
  <rdf:type rdf:resource="COARE.owl#Marker"/>
  <rdf:type>
    <owl:Restriction>
      <owl:onProperty rdf:resource="COARE.owl#ID"/>
      <owl:allValuesFrom rdf:resource="http://www.w3.org
      /2001/ XMLSchema#string"/>
    </owl:Restriction>
  </rdf:type>
  <rdf:type>
    <owl:Restriction>
      <owl:onProperty rdf:resource="COARE.owl#InView"/>
      <owl:allValuesFrom rdf:resource="http://www.w3.org
      /2001/ XMLSchema#boolean"/>
    </owl:Restriction>
  </rdf:type>
</owl:NamedIndividual>
```

(c)

Figure 5.6: Ontology representation

5.5.2.2 Logical rules definition

To support context reasoning and adaptation, SWRL (Semantic Web Rule Language) rules are used to define the logical relationships among the contexts A SWRL rule composes of a body (a set of conditions) and a head (a set of actions to be taken if all the conditions within the body are satisfied). Each condition or action in the rule is a rule atom, and there are seven types of atoms that can be used in SWRL rules [SWRL]. Within each atom, variables that refer to the instances of concepts in the ontology (class, subclass, and property) can be used in the atoms.

As an example, the rule for inferring the status of the scheduled activity (Figure 5.6(a)) is illustrated in Figure 5.7(a). In this example, the rule infers that the status of CheckOilLevel should be Start when the local time is equal to its start time. The rule in Figure 5.7(b) shows that when the marker Hiro (Figure 5.6(c)) is in the user's view, the user is regarded as located in the surroundings of the cabinet (Figure 5.6(b)).

(a)

```
ScheduledActivity(CheckOilLevel)^LocatedIn(CheckOilLevel,?a)^
StartTime(CheckOilLevel,?b) ^ Time(?a,?c) ^
swrlb:equal(?b, ?c) => Status(CheckOilLevel, START)
```

(b)

```
Marker(Hiro) ^ InView (Hiro, TRUE) => Person (?a) ^
IndoorSpace(Cabinet) ^ LocatedIn(?a, Cabinet)
```

Figure 5.7: Examples of logical rules

5.5.2.3 Context model review

The context knowledge authoring module is based on the Pellet reasoner [Pellet] and it allows the AR developers to review the consistency of the context model that has been constructed by the users (Figure 5.8). In case of errors, the unsatisfiable classes and the logical rules causing the inconsistency will be highlighted to the AR developers for revision.

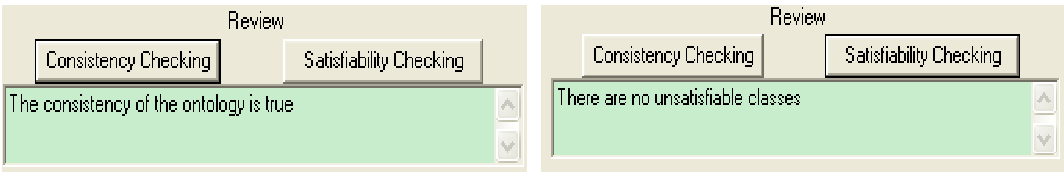


Figure 5.8: Context model review

5.5.3 Information instance authoring

Authoring the information instances involves inserting the media files, registering the information instances in the context ontology and relating them to the various contexts. The authoring process comprises of three steps, namely, (i) content definition, (ii) rendering definition, and (iii) authoring review. Content definition and rendering definition correspond to the content adaptation and rendering adaptation in the context adaption scheme (Figure 5.1).

5.5.3.1 Content definition

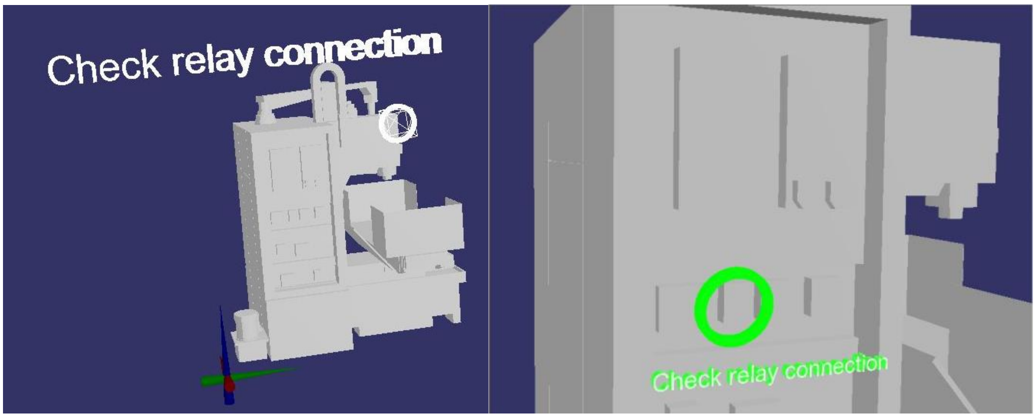
In content definition, the user can define different information instances that can be rendered in different contexts, and insert media files for these instances. An information instance can comprise a certain number of media files and has the content-related properties of item, presentation, etc. An information instance can be initialized through defining it as a subclass of the InformationInstances class in the ontology, e.g., CheckRelayConnection.

The values of the content-related properties of an instance, i.e., item and presentation, and the content adaptation rules relating the information instance to the relevant contexts should be defined. The property item can be defined based on the information described by the information instance and the property presentation should be defined according to the media files to be used to describe the information. For example, the item of CheckRelayConnection is Alert_CheckRelayConnection, and the presentation is symbolic presentation as texts and symbols will be used to describe the information. The content adaptation rules can be defined using SWRL rules as discussed in Section 5.5.2.2. For example, item should be Alert_CheckRelayConnection when the task to be conducted is AlarmLVAL at the second step and presentation should be symbolic presentation for a novice, i.e., the information instance CheckRelayConnection should be rendered (Figure 5.9).

```
ScheduledActivity(AlarmLVAL) ^ Status(AlarmLVAL, Step2) ^
Person (?a) ^ EngagedIn(?a, AlarmLVAL)^
ExpertiseLevel (?a, Low)
=> InformationInstances(CheckRelayConnection)^
Display(CheckRelayConnection, TRUE)
```

Figure 5.9: An example of content adaptation rule

A number of media files can be inserted to describe the information in an information instance. A virtual model of the application scenario can be constructed to link the real and virtual environments, and it can be displayed in the virtual scene in the user interface. A catalogue of media files, e.g., the CAD models of common mechanical tools, images and symbols, is provided for the users to use during the authoring process. The user can load any of these media files or author new virtual objects, e.g., texts, and insert them in the virtual scene. As shown in Figure 5.10(a), a circle and a series of texts have been inserted. The media files can be arranged spatially in the 3D virtual environment using computer mouse, and their visual properties, e.g., color and transparency, can be specified by the users via user interface. As shown in Figure 5.10(b), the two virtual objects have been moved to appropriate locations relative to the virtual CNC machine, and their colors have been changed to green.



(a) (b)

Figure 5.10: Inserting and arranging media files

5.5.3.2 Rendering definition

Rendering adaptation rules can be defined to adjust the rendering of the information instances according to different contexts to achieve context adaptable rendering. Each information instance has two categories of rendering properties, namely, format and registration (Section 5.2.1). To adapt a particular rendering property to certain contexts, the user can add the property in the ontology and define the adaptation method using logical rules. For example, the rule in Figure 5.11 adjusts the property color according to the user's preference.

```
InformationInstances(CheckRelayConnection) ^
Display (CheckRelayConnection, true) ^ Person (?a) ^
PreferedColor (?a, ?b)=>Color(CheckRelayConnection, ?b)
```

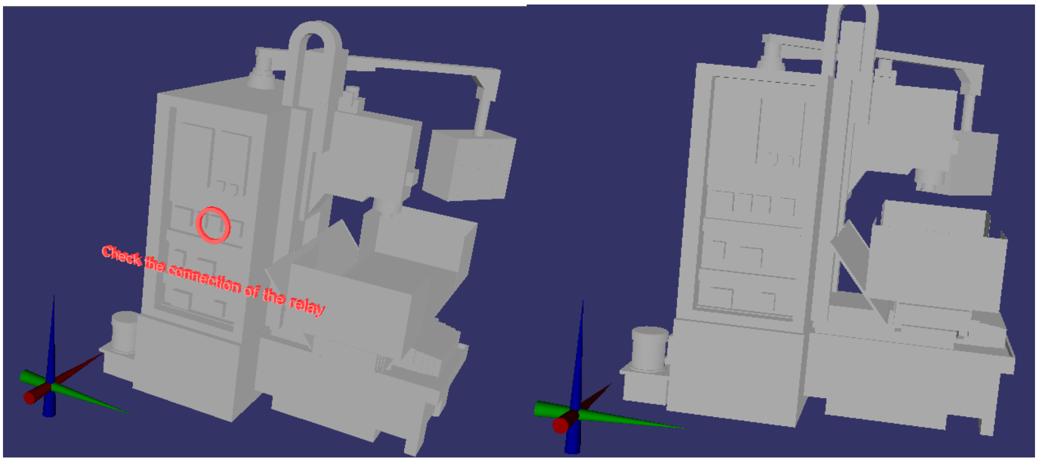
Figure 5.11: Rendering adaptation rule

5.5.3.3 Information instance review

The information instances that have been authored can be reviewed through simulating various contexts via the user interface so as to determine whether the information instances can be adapted to the contexts correctly.

In offline authoring, the users can define the contexts which will be transmitted to the context management module. The context management module will derive the high-level contexts from the simulated low-level contexts and transmit the contexts to the adaptation engine, which will adapt the information instances to different contexts. The information instances filtered in the simulated contexts will be displayed in the virtual scene for the users to review. For example, the user has authored the information instance CheckRelayConnection (Figure 5.10(b)), and it is defined to be displayed to the user with low level of expertise at Step2 of activity AlarmLVAL (Figure 5.9), and the virtual objects should be rendered in the user's preferred color (Figure 5.11). To review this authoring, the user can define the values of the properties Status as Step2, PreferedColor as Red, and ExpertiseLevel as Low, such that the information instance CheckRelayConnection will be rendered as shown in Figure 5.12(a). By defining Status as Step3, the information instance will not be rendered (Figure 5.12(b)). In case of improper rendering, the author can revise the information instance.

Finally, the context knowledge that has been modeled and information instances that have been defined can be saved and stored. The context ontology and logical rules will be stored in the database as an owl file, and a text file containing the information of the media files will be generated for each information instance. The text file records the file path of each media file and the rendering parameters, e.g., color, translation and rotation.



(a) (b)

Figure 5.12: Authoring review

5.6 On-site Authoring

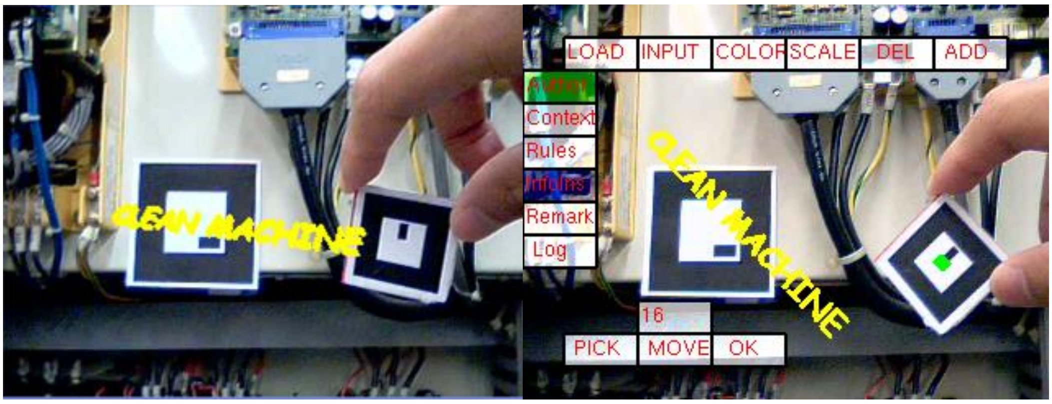
The OSA module allows the maintenance technicians to interact with and update the AR-based maintenance knowledge actively instead of just being passive information receivers. Using the OSA module, the technicians can review and edit the information instances that have been authored by the AR developers and augmented in their views, and record new maintenance information and details.

5.6.1 Mobile user interface

A mobile user interface, which consists of a physical marker and a virtual panel [Yuan *et al.* 2004], is provided to the technicians. The physical marker can be tracked in 3D space [ARToolKit 2005], and it is used to support two types of interactions, namely, 2D cursor and 3D placement tool. The image coordinates of the center of the marker is drawn as a green dot, and it is used as the 2D cursor (Figure 5.13). The translation and rotation of the marker are used to arrange the virtual objects spatially, i.e., 3D placement tool. As shown in Figure 5.14, the text message “CLEAN MACHINE” can be arranged in the 3D space according to the spatial movements of the 3D placement tool.

The virtual panel in the system is based on the screen coordinate, and it is a virtual display of computer augmented information, such as virtual buttons (Figure 5.13). Therefore, with the 2D cursor, the user can activate the virtual buttons to achieve interactions with the system. To activate a virtual button, the user can place the 2D cursor on it for a predefined time range.

Figure 5.13: 2D cursor and virtual panel



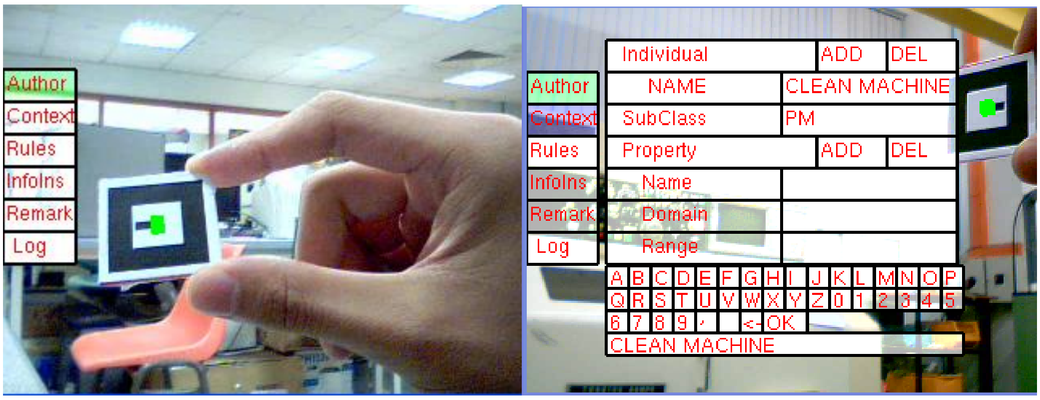
(a) (b)

Figure 5.14 3D placement tool

5.6.2 Authoring

On-site authoring is an authoring-and-review process; authoring is conducted by the maintenance technicians onsite and an authoring log recording the details of the authoring can be generated; a review will be conducted by the AR developers based on the authoring log to evaluate whether the newly authored and/or edited AR contents should be updated in the database and shared with other maintenance technicians.

During the authoring process, the maintenance technicians can edit the contents authored by the AR developers or create new information instances to update the maintenance knowledge base. To achieve this, the OSA module provides access to the context model and information instances authored by the AR developers, as well as the media files stored in the database. To edit and update the context model and the information instances, the technician can activate the corresponding button rendered on the left side of the view plane (Figure 5.15 (a)). The context model can be edited by adding or deleting instances of concepts (individuals and properties), e.g., a new preventive maintenance activity of “clean machine” can be added (Figure 5.15 (b)). The content adaptation rules can be edited through the virtual panel by relating the contexts to the property values. For example, at step five of activity CM1, Item will be CLEAN (Figure 5.15 (c)). Finally, the information instances can be edited by inserting and arranging the media files, modifying their visual properties, and applying certain rendering rules. As shown in Figure 5.15 (a), a circle has been loaded, and the technician is adding a text message via the virtual keyboard rendered.



(a) Main Panel (b) Update Context Model

(c) Add Content Adaptation Rules (d) Edit Information instance

Figure 5.15 On-site Authoring

Although the user interface abstracts the low-level programming skills, the maintenance technicians may still encounter difficulties during the authoring process. In such cases, they can describe their intention and difficulties in the remarks, e.g., they can indicate the particular contexts that the information instance authored should be filtered using texts when they have difficulties defining the content adaptation rules. These remarks will be used by the AR developers to implement the intended authoring of the technicians during the review process.

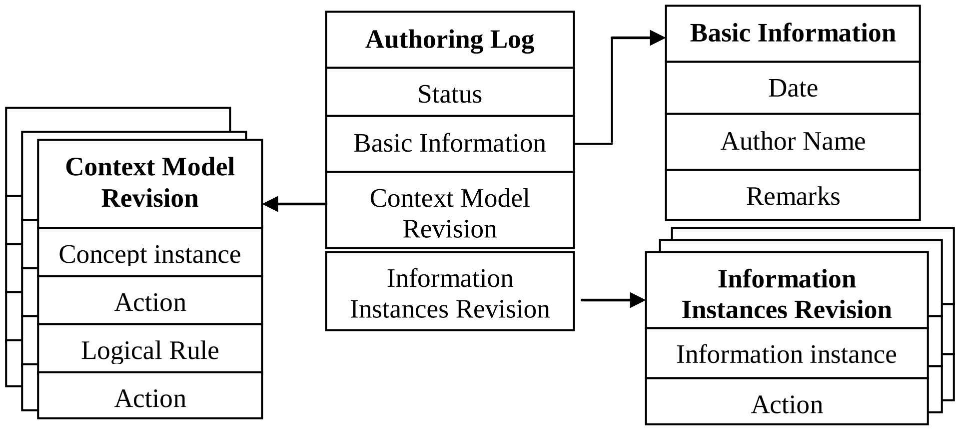


Figure 5.16 Authoring Log

Finally, an authoring log (Figure 5.16) will be generated for the AR developers' review. It comprises of four parts, namely, status, basic information, context model revision, and information instances revision. The status part is determined by the AR developers, and it can be pending, approved, or rejected. Only approved authoring will be used to update the database. The basic information part includes the authoring date, author name, and remarks. The context model revision part records the revisions on the context model, and it includes a pair of concept instance (class, subclass, and property)/logical rule and the action (add or delete) taken. According to the authoring logs, the AR developers can review the authoring using the OFA module and decide whether it should be approved.

5.7 Context-aware AR services

The CM and ARV modules collect and reason contexts, and filter, adapt and register the information instances in the real environments to provide context-aware AR services to the maintenance technicians.

5.7.1 Context management

The CM module consists of three parts, namely, context provider, context model, and context reasoner. The context provider acquires various context data and information and encodes them in the form of OWL descriptions. The contexts of the users need to be obtained and updated in real time in order to provide context-aware AR services to the users. The frequency to obtain and update contexts can be different for various contexts. For example, the StartTime of an activity can be defined as 9:00 am daily permanently, i.e., the activity is to be conducted at 9:00 am everyday. However, the location of the user needs to be tracked and updated in real time. The ARAMS system provides two modules to obtain and update the information of the contexts, namely, the OFA module and the CM module. Using the OFA module, users can define the values of certain contexts via the user interface, and the contexts that have been defined will be stored in the database.

The context provider in the CM module acquires raw data from the users and various other sources, e.g., web servers and physical sensors, and interprets the raw data to obtain low-level contexts. The context provider has individual functional modules to interact with each data source. For example, it collects the local time from the digital timer, and derives the equipment operating time through subtracting the local time by the start operating time. It has the tracking functional module which processes the real time images of the real environment captured by the camera, and outputs the desired low-level contexts in the ontology, e.g., translation and rotation of the camera, and ID of the marker in view. The context provider also has the user interface that collects and interprets the user inputs, e.g., ExpertiseLevel and PreferedColor. After the raw data from the data sources and user inputs have been collected and processed, the context provider encodes them to the form of the OWL descriptions in the ontology. For example, the ExpertiseLevel of the Person John is Low can be encoded as shown in Figure 5.17. The CM module will notify the ARV module to update the AR services whenever new contexts are collected.

```
<owl:NamedIndividual rdf:about="# John ">
  <rdf:type rdf:resource="# Person "/>
    < ExpertiseLevel rdf:datatype= "http://www.w3.org/2001/
    XMLSchema#string"> Low
  </ ExpertiseLevel >
</owl:NamedIndividual>
```

Figure 5.17: Context provider

The context reasoner derives high-level contexts from the low-level contexts and processes context knowledge queries from the ARV module. This context reasoning is based on the contexts collected in real time and the logical rules that have been defined. For example, the Status of an activity can be derived by comparing the local time and the StartTime of an activity. (Figure 5.7(a)). Pellet reasoner is employed in the system for context reasoning and query processing.

5.7.2 AR-based visualization

The ARV module queries the real time context knowledge and provides adaptable AR services, and it comprises of three parts, namely, content adaptation, rendering adaptation, and rendering. As shown in Figure 5.1, it determines the AR contents to be provided according to the current contexts and determines the rendering parameters of these AR contents. The information instances that would be displayed in the current contexts are the display property which has value of TRUE. The rendering parameters of the virtual objects can be fixed or are context-adaptable. The values of the fixed rendering parameters are obtained from the files that record the authoring of the virtual objects in the virtual scene (Section 5.5.3.3), and the values of the context-adaptable parameters are queried from the real time property values of the information instance, e.g., color (Figure 5.11). Finally, a set of rendering units, which comprises of the virtual objects and their rendering properties (Figure 5.18), will be generated and displayed in the AR environments.

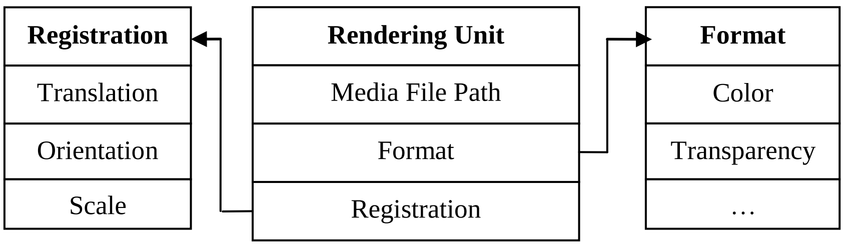


Figure 5.18: Data structure of a rendering unit

5.8 Summary

In this chapter, the concepts and techniques of context-aware AR contents authoring and visualization have been presented. A bi-directional authoring tool has been proposed to enable the AR developers to create context-aware AR contents offline and the maintenance technicians to edit and update the AR contents on-site. Intuitive user interfaces are provided to abstract the low-level programming techniques so that users with little programming skills to participate in the authoring process. In addition, the context adaptation scheme and rendering of the authored contents are described to provide context-aware AR services to the maintenance technicians.

Chapter 6 System Implementation and Evaluation

This chapter presents the development of the prototype ARAMS system based on the methodologies described in chapters 3, 4, and 5. In addition, case studies and user studies have been conducted to validate the proposed concepts and techniques, and evaluate the performance of the ARAMS system. Finally, user feedback has been collected to provide suggestions on potential future improvement of the proposed system.

OWL API [OWL API] is used for ontology construction and logical rules definition, Pellet reasoner [Pellet] is used for context reasoning and inferring context knowledge, OpenSceneGraph [OpenSceneGraph] is used to draw the virtual scene in the user interface, and ARToolKit [ARToolKit 2005] and PTAM [Klein and Murray 2007] are used for AR implementation. The CM module is developed using Java, the tracking, ARV, OSA, OFA, and ONA modules are developed in C++, and the system is integrated via socket communications.

Three user studies have been designed to evaluate the effectiveness of the using ARAMS for AR-based remote maintenance, context-aware AR contents authoring, and context-aware AR maintenance information visualization respectively. Details of each study will be described accordingly.

6.1 User Study on Using ARAMS for Remote Maintenance

This user study was designed to test the performance of the ARAMS system in remote maintenance and compare the proposed collaborative method (AR-based multimedia instructions) with the traditional collaborative method, namely, verbal instructions. Twelve researchers in the author's laboratory participated in this user study, and they are divided into two groups, i.e., six participants per group. In each group, three participants act as the experts and the other three participants act as the technicians. The two groups use the traditional verbal-based collaborative method and the proposed AR-based collaborative method respectively in this user study to test the system usability.

The experts work in an office environment equipped with a desktop, a keyboard, a mouse and an audio and micro headset. The technicians work in the remote maintenance site and wear the mobile platform. Wireless communication is established between the two environments.

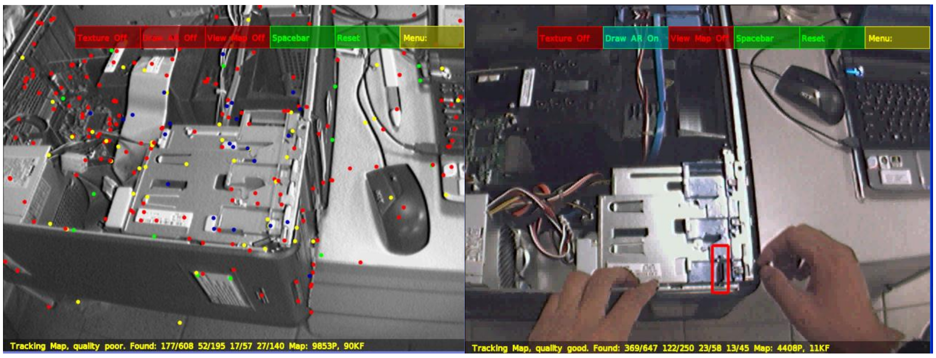
6.1.1 Tasks

In this user study, the experts instruct the technicians to disassemble the CD tray of a computer. This tele-assistance is conducted based on both the traditional verbal-based collaborative method (Group I) and the proposed AR-based collaborative method (Group II).

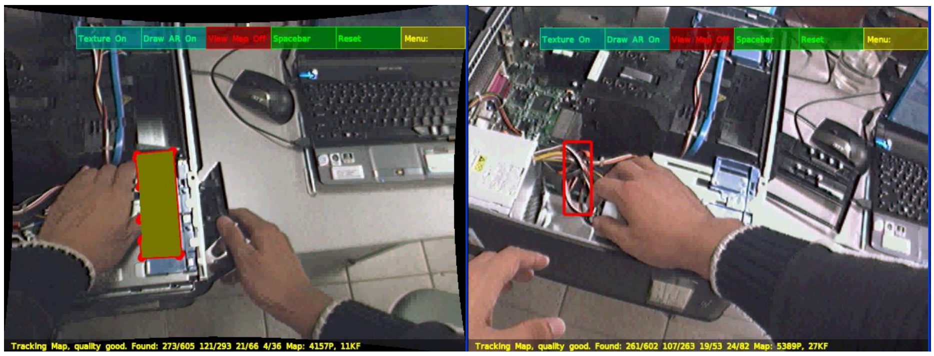
The detailed process of tele-assistance using the proposed collaborative method is as follows. The expert first instructs the technician to open the computer chassis through verbal instructions. The 3D map, which comprises of a bunch of feature points of the computer (drawn as colored dots in Figure 6.1(a)), is created using PTAM online when the technician translates and rotates the camera at different perspectives of the computer. To enable the technician to locate the spring, the expert outlines the spring by augmenting a red rectangle and asks him to unfasten it (Figure 6.1(b)). In Figure 6.1(c), the outline of the plate is modeled and its removal motion is animated. Viewing the animation augmented on the real plate, the technician can follow the operation to slide it out. The expert outlines the wires using a red rectangle and asks the technician to unplug the wires through verbal instructions (Figure 6.1(d)).

Based on the 3D map that has been created, the expert constructs a simplified model of the CD tray (Figure 6.1(e)), locates the length of the CD tray (Figure 6.1(f)), and translates the virtual model along its length. The translational motion of the CD tray is animated on top of the real component (Figures 6.1(g, h)). In Figure 6.1(i), the virtual model of the CD tray is slid out of the computer, and the technician performs this operation on the real component.

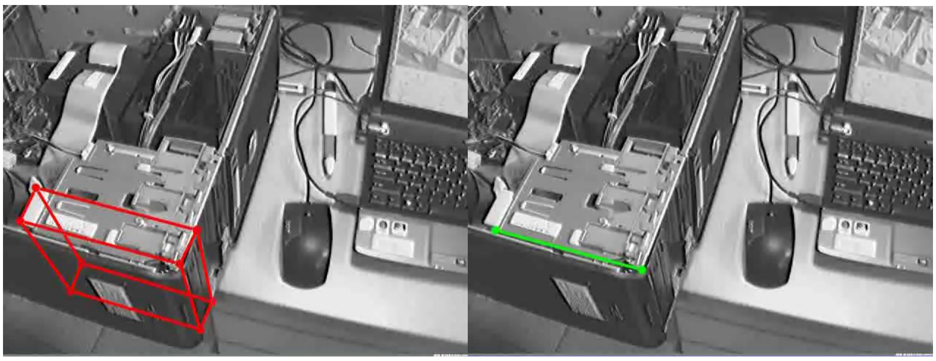
To account for the video transmission delays and avoid potential safety issues, both visual-based and verbal-based communications are used in this collaborative work. All the operations can only be conducted when both visual and verbal commands from the experts are coherent.



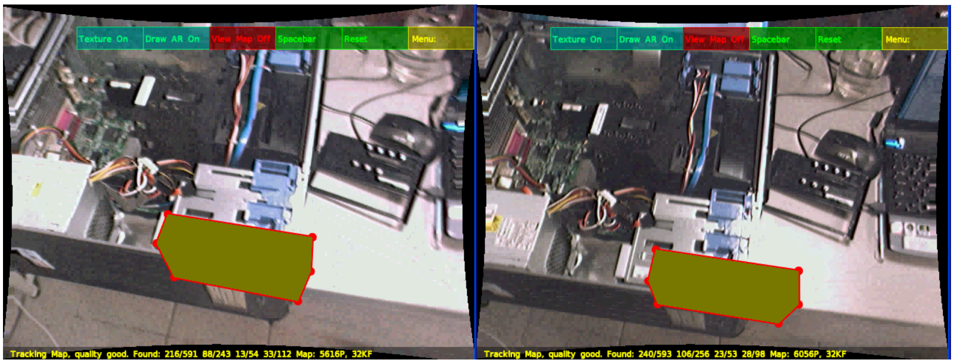
(a) Created 3D map (b) Unfasten the spring



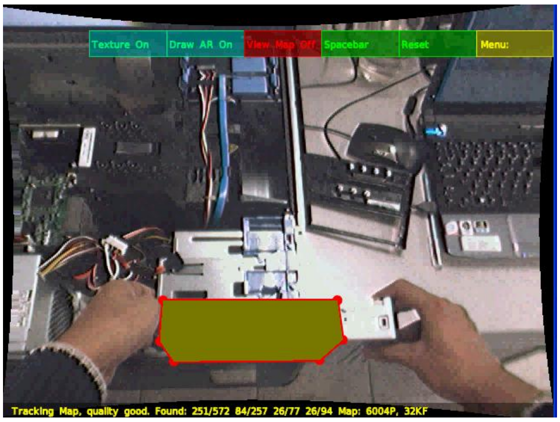
(c) Slide out the plate (d) Unplug the wires



(e) Construct simplified model of CD tray (f) Locate the length of CD tray



(g & h) Animation of the virtual CD tray translating along its length



(i) Slide out the CD tray

Figure 6.1: User study on using ARAMS for remote maintenance

6.1.2 Discussion

Both quantitative and qualitative data have been recorded to analyze the usability of the ARAMS system and compare the proposed collaborative approach with the traditional verbal-based collaboration method (Appendix A). The video transmission delay is around 1 to 2 seconds in this study, and using both visual-based and verbal-based communications can successfully avoid potential wrong operations. The average completion time of each step and the total number of errors made by the technician are shown in Table 6.1. The errors in this user study are defined as instances when a technician locates the wrong component or performs the wrong operation.

The user study indicates that the proposed system has enabled more efficient and less error prone remote collaboration. However, as the 2D tracking method employed was not sufficiently robust, the rendered rectangle moved around in the view of the technician in Step two, and this has resulted in a longer operation time and more errors.

In this user study, each participant is required to complete a post-experiment questionnaire, which features the five-point Likert scale questions (1=most negative, 5=most positive) to evaluate the intuitiveness, satisfaction level, and ease of use of the traditional collaborative method and the proposed method supported by the system (Table 6.2). The result shows that the proposed collaborative method is more intuitive and satisfying to the participants. The participants also provided a few suggestions and comments on the prototype system, such as “It would be better to demonstrate the maintenance process using virtual models with textures”, “The modeling method cannot model occluded parts of the components”, and “The tracking and registration accuracy should be improved”.

Table 6.1: Quantitative analysis

Parameter / Step	Average completion time/(s)		Total number of errors	
	Traditional method	Proposed method	Traditional method	Proposed method
Open the computer chassis	18	18	0	0
Unfasten the spring	21	24	0	2
Slide out the plate	16	16	1	0
Unplug the wires	38	21	2	0
Slide out the CD tray	14	12	0	0
Total	107	91	3	2

Table 6.2: Qualitative analysis

Method / Parameter	Expert		Technician	
	Traditional method	Proposed method	Traditional method	Proposed method
Intuitiveness (Average)	2.75	4	2.75	4.25
Satisfaction (Average)	3	3.25	2.75	3.75
Ease of use (Average)	3	3.5	3	4

6.2 User Study on Using ARAMS for Context-aware AR Contents Authoring

An important functional feature of the ARMAS system is allowing the AR developers to author context-aware AR contents. The objective of this user study is to evaluate the performance of this authoring tool, validate the proposed concepts and techniques of context-aware AR contents, and collect user feedback.

6.2.1 Tasks

The ARMAS system provides an offline authoring tool for developing context-aware AR contents. The system allows the AR developers to insert virtual objects in AR scenes and author their rendering properties. In addition, it explores the logical aspects of AR contents and allows the AR developers to relate the properties of AR contents to the contexts. To validate the performance of this authoring tool, a user study is conducted. This user study consists of two phases, namely, (i) benchmark the proposed tool with other AR authoring tools in terms of constructing an AR scene through adding and arranging media files, and (ii) obtain user feedback on the logical aspect of AR authoring using this tool.

In Phase I, two AR authoring tools, namely, BuildAR [BuildAR] and iaTAR [Lee and Kim 2009] have been set up. BuildAR provides a 2D user interface to allow users to author an AR scene using a marker (Figure 6.2(a)). iaTAR provides an immersive user interface to allow users to create AR contents on-site (Figure 6.2(b)). Six graduate students with experience in developing AR applications participated in this user study. These participants learned the use of three authoring tools and used these tools to author an AR scene that describes the configuration of a CNC machine (Figure 6.2(b)). During the authoring process, the participants can load texts, symbols and 3D models, arrange them spatially and modify their rendering properties. The average authoring times using these three tools are shown in Table 6.3.

In Phase II, the participants used the ARAMS system to conduct authoring for a context-aware AR application that will be used to assist the machine operators and maintenance engineers in the use and maintenance of a CNC milling machine in the manufacturing laboratory of the author’s institution. In this study, information on the equipment configuration and maintenance instructions is to be authored. The CAD model of the CNC milling machine is constructed to provide 3D positional references during the authoring process, and two ARToolKit markers, Hiro and Kanji, are attached on the equipment for AR implementation. The contexts of the application environment are modeled (Figure 6.3), and each participant uses the proposed authoring tool to construct the ontology and author two information instances.

(a) BuildAR (b) iaTAR

Figure 6.2: System setup

Table 6.3: Task completion time

	ARAMS	BuildAR	iaTAR
Average completion time	15m 10s	15m 34s	9m 8s

In the ontology, the Person class has two subclasses, namely, the Engineer who maintains the equipment and the Operator that uses the equipment. The Person class has several properties, e.g., Name, ExpertiseLevel and PreferedColor. The Activity class has subclass LVALAlarm and a property of Status. LVALAlarm is a corrective maintenance activity for the CNC milling machine. The class Location has two subclasses, namely, Workspace and Cabinet. The subclass Workspace is the front region of the machine, and the subclass Cabinet is the right side of the machine. Three physical sensors, namely, a camera and two markers are registered in the class SensorDevice, and the properties of Translation and Rotation are used to store their positional information. The laptop of the mobile platform is registered in the class CompDevice, and the CNCMillingMachine is a subclass of the class Equipment. There are ten information instances registered, and they have properties of display, item, presentation and color.

The participants are briefed on the knowledge of context modeling and the usage of the proposed system in advance. In addition, instructions and assistance are provided to the participants during the authoring processes. The ontology is designed and provided to the participants, and the participants use the proposed tool to construct it. In addition, assistance is provided to the participants on forming SWRL rules when they cannot utilize the proper rule atoms to express the logic for more than one minute. Although assistance on the design of the context model is required by several participants, little difficulty is encountered on using the ARAMS system as a supporting development tool to author the context model and information instances. All the participants complete the task correctly.

After the authoring process, qualitative data are collected to evaluate whether the proposed tool is easy to learn and use. Each participant has to evaluate the ease of learning and ease of use of the three authoring tools using five-point Likert-based questions (1 = most negative, 5 = most positive). In addition, the participants are required to provide feedback on the pros and cons of each authoring tool, and the obtained qualitative results are shown in Table 6.4.

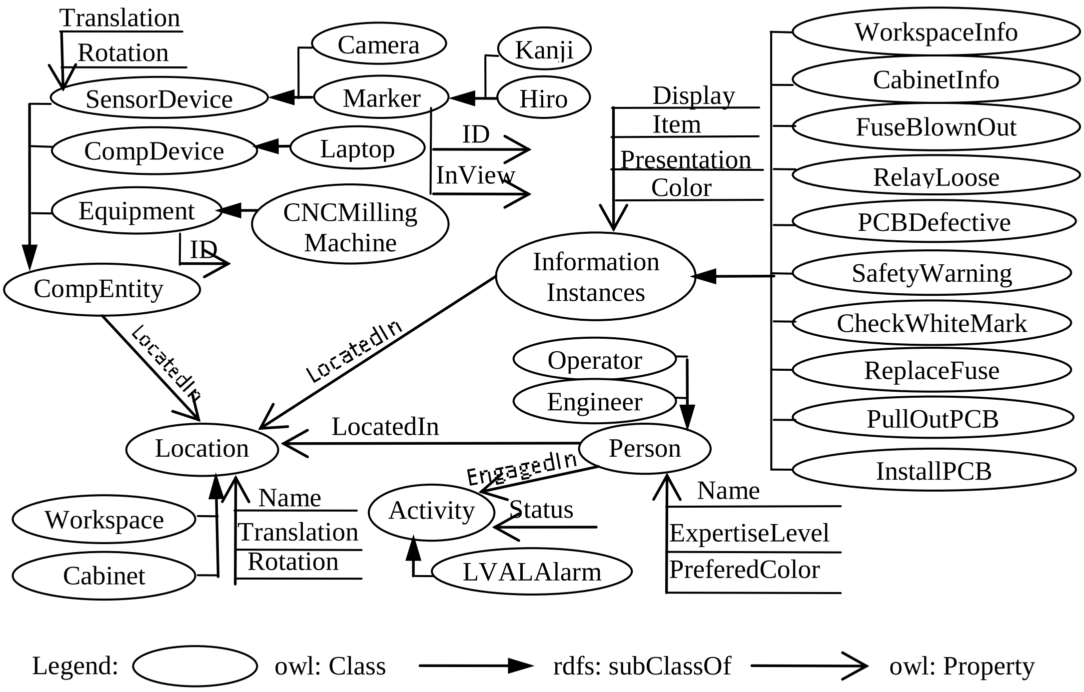


Figure 6.3: Context ontology

Table 6.4: Qualitative analysis

		ARAMS	BuildAR	iaTAR
Ease of learning		3.5	4	4
Ease of use		3.5	4	3.5
Usability	Pros	Functions for context-awareness	Easy and cheap system setup	Immersive user interface
	Cons	Not immersive authoring	Limited functions	Bulky platform

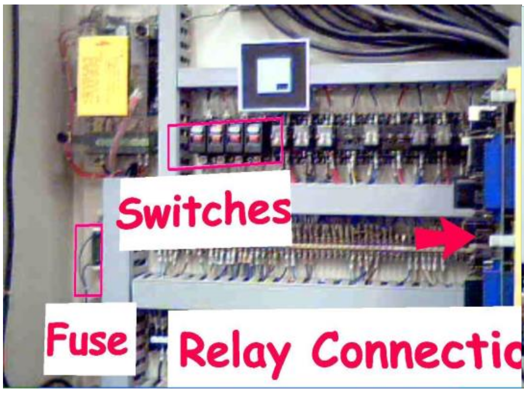
6.2.2 On-site testing

On-site testing is conducted to review the authored information instances on-site and test whether the context-aware AR contents can be adapted correctly. The real time positional information of the camera and the marker in view are obtained through processing the images captured using ARToolKit. The location of the user is derived from the ID of the marker in view, i.e., the location is Workspace if Hiro is in his view and the location is Cabinet if Kanji is in his view. The contexts of

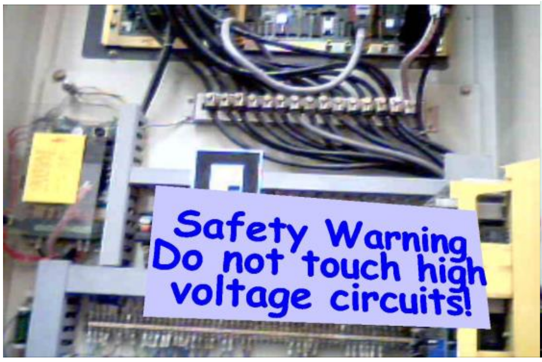
ExpertiseLevel and PreferedColor of the user, and Status of the activity LVALAlarm are collected from user inputs via the mobile user interface.

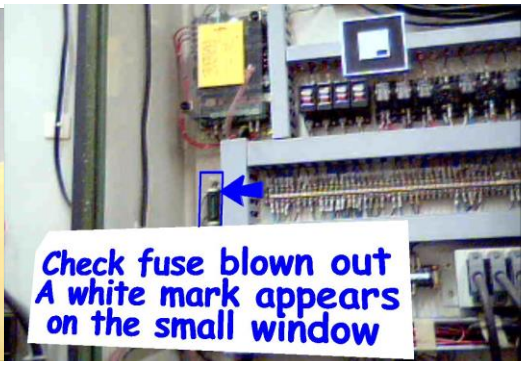
When a user logs in the system as the operator, the configuration information of the CNC milling machine will be provided to him. Information instances WorkspaceInfo and CabinetInfo, which describe the information of the equipment workspace and cabinet respectively, will be provided to the operator based on his location and rendered in his preferred color (Figures 6.2(b), 6.4(a)), e.g., WorkspaceInfo will be provided when the operator is located in Workspace.

When the user logs in the system as an engineer, selects his expertise level and preferred color, and the instructions on the activity LVALAlarm will be provided. Information instances FuseBlownOut, RelayLoose and PCBDefective are instructions on the three steps in the activity LVALAlarm, and they will be provided to experienced engineers at each step. The other five information instances are of high level of detail describing the operations of these three steps. Information instances SafetyWarning, CheckWhiteMark, and ReplaceFuse describe the three sub-operations of FuseBlownOut, and information instances PullOutPCB and InstallPCB describe the two sub-operations of PCBDefective. These detailed instructions will be provided to less experienced engineers. For example, information instance FuseBlownOut outlines the fuse to the experienced engineer (Figure 6.4(b)), and the information instances SafetyWarning, CheckWhiteMark and ReplaceFuse describe the detailed operations to the less experienced engineers (Figure 6.4(c-e)).

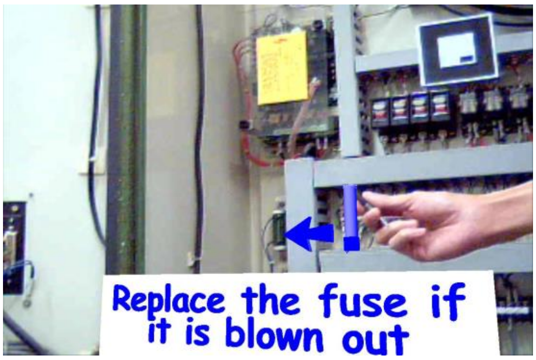


(a) CabinetInfo (b) FuseBlownOut





(c) SafetyWarning (d) CheckWhiteMark



(e) ReplaceFuse

Figure 6.4: Context-aware AR services

6.2.3 Discussion

Both quantitative and qualitative data are recorded in this study (Appendix B). The average completion times of using the three authoring tools (Table 6.3) show that the ARAMS system provides comparable functions for constructing AR scenes through loading and arranging virtual objects. iaTAR has the best performance in Phase I due to its 3D immersive user interface that facilitates the spatial arrangement of the virtual objects. However, the 3D placement tool (marker) and virtual buttons in the immersive user interface may not be suitable for logical tasks compared to mouse and keyboard [Lee and Kim 2009]. The qualitative feedback in Table 6.4 shows that the ARAMS system is easy to use and features authoring context-aware AR contents. The ARAMS system is rated lower in ease of learning as it has many more functions compared to the other two tools, e.g., context modeling and logical rules definition, and the participants' lack of such prior knowledge and experiences further complicate the learning and authoring process. However, the authoring is completed successfully when instructions are provided, and the participants indicated that the proposed authoring tool is easy to use when they have become familiar with the system.

Finally, the participants provided comments on the usability of each tool (Table 6.4), and all the participants felt that authoring context-aware AR contents is important and the proposed authoring tool is useful for developing such AR applications. Several participants commented that the authoring experience is not immersive compared to the other two tools, as the authoring is based on the 2D desktop user interface and relative to the CAD model of the application environment. They suggested that the system should provide a flexible user interface which combines both 3D immersive interface for spatial tasks and 2D desktop-based interface for logical tasks, etc. BuildAR is easiest to learn and use, and features easy and cheap system setup, but provides very simple and limited functions. Thus, it is more suitable for people with little knowledge of AR to create simple AR applications. iaTAR immerses the users in the AR scenes and allows easy spatial arrangement of the virtual objects, but the authoring platform comprising of HMD, computer, etc., is bulky and not easy to wear.

In the on-site test, it was found that the authored information instances can be adapted to the contexts correctly. The information instances with the right content can be filtered according to the user's location, role, and expertise level and task status, and the virtual objects are rendered in the user's preferred color in dynamic adaptation. However, it was found that the rendered virtual objects jitter when the user moves further away from the marker and disappears when the marker is not in the view. Other tracking methods based on natural features or virtual models that provide more stable tracking results will be explored in future.

6.3 User Study on Using ARAMS for Context-aware AR Contents Visualization

Providing context-aware AR-based information to assist the maintenance technicians is one of the main objectives of this research. This user study is to evaluate the effectiveness of using the ARAMS system to assist the technicians compared to the traditional paper manuals. In addition, the functional module of on-site authoring will be evaluated by the participants.

6.3.1 Application scenario

The CNC milling machine in the manufacturing laboratory of the author's institution (Section 6.2) is used in the user study to assess the performance of the prototype system. Context-aware AR information will be provided to the various users of the equipment:

- Instructions on maintaining the equipment will be provided to the engineers according to the task status and the engineers' expertise level;
- The virtual objects will be rendered in the users' preferred colors; and
- Maintenance information can be interacted with and authored by the engineers on-site.

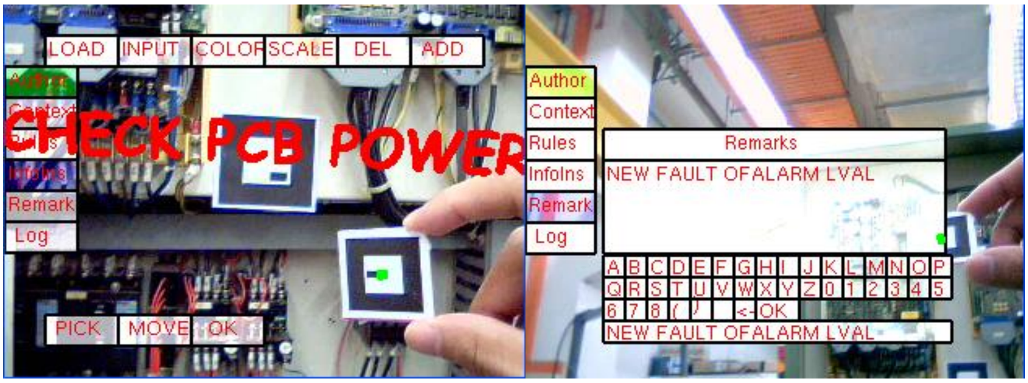
6.3.2 Offline authoring

The context model and information instances are authored as described in Section 6.2.1, and they will be used for assisting the participants in this user study.

6.3.3 Tasks

There are eight participants in this case study and they are divided into two groups, namely (I) four researchers in the author's laboratory with AR knowledge, and (II) another four researchers in the author's laboratory also with AR knowledge.

The user study is conducted as follows. Groups I and II participants are instructed to conduct the task LVALAlarm twice. For the first time, all of them do not know the task procedures and they are regarded as having low level of expertise. Paper-based manuals are provided to the participants in group I and the proposed system is used to assist the participants in group II. High level-of-detail instructions are provided to the group II participants due to their low level of expertise (Figure 6.4(c–e)). For the second time, the participants are regarded having high level of expertise as they have been familiarized with the task. The participants in group I are provided with traditional AR-assisted maintenance systems, i.e., the instructions provided to all users are the same regardless of their expertise level (Figure 6.4(c–e)) and cannot be authored. The participants in group II use the proposed system, and they are provided with low level-of-detail instructions according to their expertise level (Figure 6.4(b)). Finally, the group II participants are encouraged to add a new potential fault of the failure by authoring an information instance, CHECK PCB POWER (Figure 6.5(a)), and indicating the desired rendering contexts in remarks (Figure 6.5(b)).



(a) Add a new fault (b) Indicate the desired contexts

Figure 6.5: On-site authoring

6.3.4 Discussion

Both quantitative and qualitative data have been recorded to evaluate the performance and usability of the system and compare the proposed system with paper-based manuals and the traditional AR-assisted maintenance systems (Appendix C). The average completion time for each step in LVALAlarm is shown in Table 6.5. The results show that traditional AR-assisted maintenance systems can improve the workflow as compared to the paper-based manuals, and the proposed system can further enhance the efficiency through providing context-aware information. In addition, each participant is required to complete a questionnaire that contains the five point Likert questions (5 0 most positive, 1 0 most negative) to evaluate the intuitiveness, ease of use, and satisfaction level of the instructions provided (Table 6.6). The result shows that the proposed system is more intuitive, satisfying, and easier to use as compared to paper-based manuals and traditional AR-assisted maintenance systems.

Finally, several suggestions and comments are provided by the participants. A participant from group II suggested that “The tracking and registration accuracy need to be improved, as the rendered information jitters around” and “Authoring the rendered information on-site is quite useful, but it is not easy to arrange the virtual objects in 3D space using the planar marker. More intuitive interaction tools should be employed.”

Table 6.5: Quantitative analysis

Parameter \ Step			FuseBlownOut	RelayLoose	PCBDefective	Total
Average completion time /(s)	First time	Group I	130	123	92	345
		Group II	102	90	75	267
	Second time	Group I	97	89	70	256
		Group II	61	85	60	206

Table 6.6: Qualitative analysis

Parameter \ Step	Intuitiveness (Average)	Satisfaction (Average)	Ease of use (Average)
Paper-based Manuals	2.75	3	3
Traditional AR-assisted maintenance system	3.75	3.5	3.25
Proposed system	4.25	4.5	3.75

6.4 Summary

The prototype ARAMS system has been developed based on the proposed methodologies. Three user studies have been conducted to validate the concepts and techniques presented in this research and evaluate the system performance. These studies indicate that proposed system can improve maintenance workflow and enhance equipment serviceability. Context-aware AR information can be created easily and augmented in the maintenance environments, technicians can record and share the maintenance knowledge and experiences through authoring AR-based contents on-site, and remote experts can create and use AR-based visual interactions in remote maintenance. The prototype system has achieved the research objectives, and can assist the maintenance technicians effectively.

In addition, a few limitations of the prototype system are revealed in these studies. In remote maintenance, occlusions of the objects pose a challenge to the CV-based modeling approach. The experts cannot model objects that are invisible in the keyframes. The proposed authoring tool for developing context-aware AR contents is based on CAD models and markers. However, the CAD models are usually not available in advance, and it can be time-consuming to construct them. It is also inconvenient to attach the markers in the maintenance environment and on the equipment, and safety regulations may hinder such modifications on the maintenance environments. In addition, it is not very intuitive for the maintenance technicians to arrange the virtual objects in 3D space using the planar marker. Finally, it was found that the rendered virtual objects jitter when the user moves further away from the marker and disappears when the marker is not in the view.

Chapter 7 Conclusions and Recommendations

The main objective of this research is to explore the opportunities of applying AR technology to equipment maintenance. To achieve this, the reported AR-assisted maintenance systems are reviewed, and the research issues are identified. In addition, an AR-assisted maintenance system have been designed and developed to assist the technicians in (i) providing context-aware information to the technicians and allowing the technicians to author AR contents on-site, and (ii) supporting easy and effective remote maintenance using AR technology.

7.1 Contributions

Several contributions have been achieved in this research.

7.1.1 Better understanding of AR-assisted maintenance systems

In this research, the state-of-the-art AR assisted maintenance systems are reviewed in terms of the display technologies, tracking technologies, data management, human-system interaction, and remote collaboration. The current research status and potential research directions are identified. This review is useful for the researchers to understand the benefits and issues of applying AR to equipment maintenance, and improve the usability of the reported AR-assisted maintenance systems in routine and *ad hoc* maintenance environments.

7.1.2 Better understanding of AR authoring

Authoring has been neglected by most researchers although it plays an important role in the development of AR applications [Zhou *et al.* 2008]. Relatively fewer works have been reported on authoring compared to the other topics of AR, e.g., tracking and interaction. This research reviews the reported AR authoring systems and identifies the open research issues. This review highlights the importance of authoring, presents the state-of-the-art development of the AR authoring systems, and points out the potential research directions.

7.1.3 A novel online authoring tool for remote maintenance

The ARAMS system proposes methodologies to apply AR technology to the remote collaborative work on equipment maintenance. It has three main features, namely, (i) it allows the expert to create AR-based maintenance instructions online while most reported authoring systems only support offline authoring; (ii) it requires no *a priori* knowledge of the maintenance area and can be used in unprepared environments; and (iii) DWUI is proposed to improve authoring stability. The conducted user study indicates that the ARAMS system can allow the expert to create AR-based instructions efficiently and facilitate the remote maintenance processes.

7.1.4 A novel tool for authoring context-aware AR information

The ARAMS system provides a novel bi-directional authoring tool for the AR developers and the maintenance technicians to create context-aware AR contents. The concepts and properties of context-aware AR contents are explored, and intuitive user interfaces are designed to abstract the low-level programming skills from the users. In addition, the adaptation scheme of context aware AR contents is proposed to adapt the authored contents to the real time contexts and register them in the real environment.

7.1.5 A novel context-aware AR contents visualization system

This research proposes a novel AR-assisted maintenance system, which renders context-aware information using AR to assist the maintenance technicians. It aims to improve the usability of the state-of-the-art AR-assisted maintenance systems in routine and *ad hoc* maintenance activities. In ARAMS, the information that is relevant to the user's contexts will be provided. The proposed system has been proven to improve the workflow of the maintenance activities in the user study.

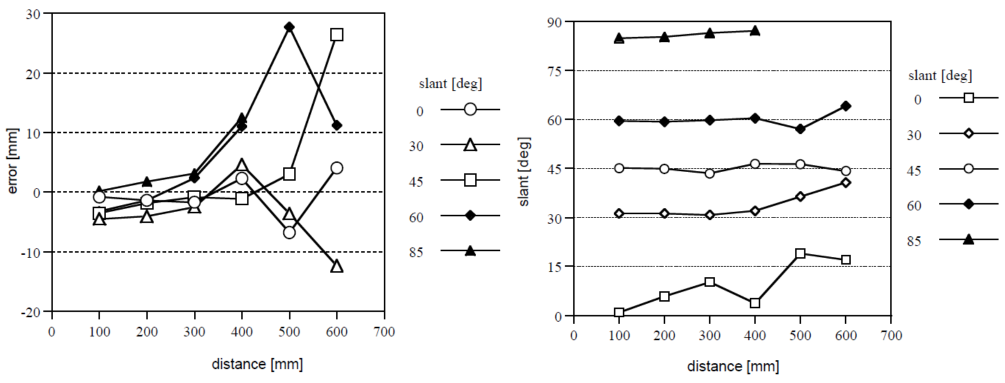
7.2 Recommendations

Despite the contributions achieved, several areas can be explored to further enhance this research work.

7.2.1 Improve the tracking accuracy

Two state-of-the-art tracking libraries, ARToolkit and PTAM, are employed in the ARAMS system to track the environment. A marker with 80 mm of side length is used to evaluate the accuracy of ARToolkit, and the errors of detected position and slant of the marker are shown in Figure 7.1. It can be seen that the tracking accuracy decreases the further the marker is away from the camera. To evaluate the accuracy of PTAM, the calculated camera trajectory is compared with the ground truth and that obtained from the EKF-SLAM algorithm [Williams et al. 2007]. A camera is moved along two textured walls at right angles, moving sideways along one wall to the corner and then along the next wall, and the camera travels 18.2m in total. As shown in Figure 7.2, the error is predominantly in direction Z, and the standard deviation from ground truth is 6mm for PTAM.

Although these two libraries provide relatively good tracking performance, the tracking accuracy still can be improved. Incorrect tracking and augmentation can potentially lead to dangerous situations, e.g., misguiding the technician to press the wrong button on the equipment. Thus, it is recommended that the tracking module of ARAMS can be improved in future.



(a) Error of position (b) Detected slant

Figure 7.1 Tracking performance of ARToolkit [ARToolKit 2005]

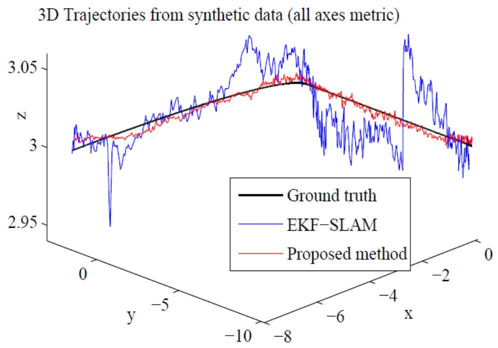


Figure 7.2 Tracking performance of PTAM [Klein and Murray 2007]

7.2.2 Decrease video transmission delays

The video transmission delay in the user study on AR-based remote maintenance is low, but such issue should not be ignored when the network has low bandwidth. High delays can decrease the efficiency of the remote collaboration and lead to potential safety issues. Related technologies, e.g., video compression, can be explored in future.

7.2.3 Remove the requirement of environment preparation

Using the ARAMS system to authoring context-aware AR contents requires *a priori* knowledge of the maintenance environment, e.g., CAD models of the equipment need to be constructed in advance. Such preparation can be time-consuming, costly, and can affect the system usability in *ad hoc* maintenance activities. Thus, the proposed system can be improved through allowing the AR developers to author contents in the unknown environments, and the potential solutions can be employing markerless tracking methods, etc.

7.2.4 Improve the mobile user interface for on-site authoring

The current mobile user interface provided to the technician is composed of a 2D marker and virtual menu. The participants in the conducted case study indicated that it is not intuitive to arrange the virtual media files spatially using the 2D marker and suggest better interaction tools should be provided. Bare-hand interaction [Shen *et al.* 2011] is a promising solution to this issue, and this method can track the user’s hand in the 3D space and thus allow the user to interact with the virtual objects intuitively using bare hands. In future, bare-hand tracking in 3D space can be explored.

7.2.5 Collect feedback from maintenance technicians

Although several user studies have been conducted in this research to test the usability of the developed systems and collect user feedback. To further test the performance of the proposed systems in routine and *ad hoc* maintenance activities, more user studies need to be conducted in various maintenance scenarios, e.g., factories, and suggestions from the maintenance technicians should be collected.

Publications from this Research

1) Zhu, J., Ong, S.K., and Nee, A.Y.C., 2011. Online authoring for augmented reality remote maintenance, *12th IASTED International Conference on Computer Graphics and Imaging*, Innsbruck, Austria, pp. 87-94.
2) Zhu, J., Ong, S.K., and Nee, A.Y.C., 2013. An authorable context-aware augmented reality system to assist the maintenance technicians, *International Journal of Advanced Manufacturing Technology*, Vol. 66, No. 9, 1699-1714.
3) Ong, S.K. and Zhu, J., 2013. A novel maintenance system for equipment serviceability improvement, *Annals of CIRP*, Vol. 62, No. 1, 39-42.
4) Zhu, J., Ong, S.K., and Nee, A.Y.C., 2013. A context-aware augmented reality system to assist the maintenance operators, *International Journal on Interactive Design and Manufacturing*. (first revision)
5) Zhu, J., Ong, S.K., and Nee, A.Y.C., 2013. A Context-aware Augmented Reality Assisted Maintenance System, *International Journal of Computer Integrated Manufacturing*. (under review)

References

[Abeykoon *et al.* 2012] Abeykoon, H.A., Karunanayaka, K.T., Kumarasinghe, J.P., Roth, G., Fernando, O.N.N., Cheok, A.D., Command Center Authoring Tool to Supervise Augmented Reality, *IEEE Virtual Reality*, pp. 99-100, Costa Mesa, CA, USA, 2012

[Alvarez *et al.* 2011] Alvarez, H., Aguinaga, I., Borro, D., Providing guidance for maintenance operations using automatic markerless augmented reality system, *IEEE International Symposium on Mixed and Augmented Reality*, Basel, Switzerland, pp. 181-90, 2011

[ARTESAS] ARTESAS—advanced augmented reality technologies for industrial service applications http://www.artesas.de [last access date: 24th, September, 2012].

[ARToolKit 2005] HITLabNZ ARToolKit 2005, http://www.artoolkit.sourceforge.net/ [last access date: 24th, September, 2012].

[Aydın *et al.* 2012] Aydın, B., Gensel, J., Calabretto, S., Tellez, B., ARCAMA-3D – A Context-Aware Augmented Reality Mobile Platform for Environmental Discovery, *International conference on Web and Wireless Geographical Information Systems*, pp. 17-26, Naples, Italy, 2012.

[Azuma 1997] Azuma, R. T., 1997. A survey of augmented reality. *Presence: Teleoperators and Virtual Environment*, vol. 6, no. 4, pp. 355-385.

[Bauer *et al.* 2001] Bauer, M., Bruegge, B., Klinker, G., MacWilliams, A., Reicher, T., Riss, S., Sandor, C., Wagner, M., Design of a Component-Based Augmented Reality Framework, *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 45-54, New York, 2001.

[Bimber and Raskar 2005] Bimber, O., Raskar, R., 2005. Spatial Augmented Reality: Merging Real and Virtual Worlds, A. K. Peters, Ltd.

[BMW Augmented Reality] BMW Augmented Reality, http://www.bmw.com/com/en/owners/service/ augmented_reality_introduction_1.html, [last access date: 24th, September, 2012].

[BuildAR] HITLabNZ, BuildAR, https://buildar.com/home#p=/, [last access date: 24th, September, 2012].

[Byun and Cheverst 2004] Byun, H.E., Cheverst, K., 2004, Utilizing context history to provide dynamic adaptations. *Applied Artificial Intelligence*, vol. 18, no. 6, pp. 533–548

[Caponio *et al.* 2011] Caponio, A., Hincapie, M., Gonzalez Mendivil, E., lMAR: Highly Parallel Architecture for Markerless Augmented Reality in Aircraft Maintenance, *International Conference on Virtual and Mixed Reality*, Orlando, FL, USA, pp. 20-9, 2011

[Coelho *et al.* 2004] Coelho, E.M., MacIntyre, B., Julier, S.J., OSGAR: A scene Graph with Uncertain Transformations, *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 6–15, Arlington, VA, USA, 2004.

[Crowder *et al.* 2005] Crowder, R., Wills, G. Hall, W., 2005. Hypermedia maintenance support applications: Benefits and development costs, *Computers in Industry*, vol. 56, no. 7, pp. 681-698.

[Dangelmaier *et al.* 2005] Dangelmaier, W., Fischer, M., Gausemeier, J., Grafe, M., Matysczok, C., Mueck, B., 2005. Virtual and augmented reality support for discrete manufacturing system simulation, *Computers in Industry*, vol. 56, no. 4, pp. 371-383.

[Demiris *et al.* 2005] Demiris, A.M., Vlahakis, V., Makri, A., Papaioannou, M., Ioannidis, N., intGuide: A platform for context-aware services featuring augmented-reality, based on the outcome of European Research Projects, *Signal Processing: Image Communication*, vol. 20, no. 9–10, pp. 927–946, 2005.

[Dey 2001] Dey, A.K., 2001, Understanding and using context. *Personal and Ubiquitous Computing*, vol. 5, no. 1, pp. 4–7

[Didier and Roussel 2005] Didier, J.Y., Roussel, D., 2005, AMRA: augmented reality assistance for train maintenance tasks. *IEEE and ACM International Symposium on Mixed and Augmented Reality: Workshop on Industrial Augmented Reality*, pp. xvii–xviii

[Fantini *et al.* 2011] Fantini, M., Persiani, F., Stefano, L. Di, Azzari, P., Salti, S., 2011. Augmented Reality for Aircraft Maintenance Training and Operations Support, *Computer Graphics and Applications, IEEE*, vol. 31, no. 1, pp. 96 – 101

[Feiner *et al.* 1993] Feiner, S., Macintyre, B., and Seligmann, D. 1993. Knowledge-based Augmented Reality, *Communications of the ACM*, vol. 36, pp. 53-62.

[Fiala 2005] Fiala, M.L., 2005. ARTag, a Fiducial Marker System Using Digital Techniques, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington, DC, vol. 2, pp. 590-596.

[Friedrich 2002] Friedrich, W., 2002. ARVIKA-augmented reality for development, production and service. *IEEE and ACM International Symposium on Mixed and Augmented Reality*, Darmstadt, Germany, pp. 3–4

[Goose *et al.* 2003] Goose, S., Sudarsky, S., Zhang, X., Navab, N., 2003. Speech-enabled augmented reality supporting mobile industrial maintenance, *Pervasive Computing, IEEE*, vol. 2, no. 1, pp. 65–70

[Grimm *et al.* 2002] Grimm, P., Haller, M., Paelke, V., Reinhold, S., Reimann, C., Zauner, J., 2002. AMIRE—Authoring Mixed Reality, *The First IEEE International Augmented Reality Toolkit Workshop*, Darmstadt, Germany, 2002, pp. 2 pp.

[Güven and Feiner 2003] Güven, S., Feiner, S., 2003. A hypermedia authoring tool for augmented and virtual reality, *New Review of Hypermedia and Multimedia*, vol. 9, pp. 89-116.

[Ha *et al.* 2010] Ha, T., Woo, W., Lee, Y., Lee, J., Ryu, J., Choi, H., Lee, K., ARtalet Tangible User Interface Based Immersive Augmented Reality Authoring Tool For Digilog Book, *International Symposium on Ubiquitous Virtual Reality*, pp. 40-43, Gwangju, South Korea, 2010.

[Haringer and Regenbrecht 2002] Haringer, M., Regenbrecht, H.T., 2002. A Pragmatic Approach to Augmented Reality Authoring, *IEEE and ACM International Symposium on Mixed and Augmented Reality*, Washington, DC, pp. 237-245.

[Harmo *et al.* 2001] Harmo, P., Halme, A., Virekoski, P., Halinen, M., Pitkänen, H., 2001, Etälä - Virtual Reality Assisted Telepresence System for Remote Maintenance, *IFAC Conference on Telematics Applications in Automation and Robotics, Weingarten*, Germany, pp. 523--528.

[Heipke 1996] Heipke, C., 1996. Overview of Image Matching Techniques. *OEEPE Workshop on the Application of Digital Photogrammetric Workstations*, Lausanne, Switzerland, pp.173-189.

[Henderson and Feiner 2009] Henderson, S.J., Feiner, S., 2009. Evaluating the Benefits of Augmented Reality for Task Localization in Maintenance of an Armored Personnel Carrier Turret, *IEEE and ACM International Symposium on Mixed and Augmented Reality*, Orlando, Florida, USA, pp.135-144.

[Henderson and Feiner 2010] Henderson, S.J., Feiner, S., 2010. Opportunistic Tangible User Interfaces for Augmented Reality, *IEEE Transactions on Visualization and Computer Graphics*, January/February 2010, vol. 16, no. 1, pp. 4-16.

[Hengel *et al.* 2009] Hengel, A.V.D., Hill, R., Ward, B., Dick, A., 2009. In Situ Image-based Modeling, *IEEE and ACM International Symposium on Mixed and Augmented Reality*, Orlando, Florida, USA, pp. 107-110.

[Jacob 1993] Jacob, R.J.K., 1993. Eye-movement-based human-computer interaction techniques: Toward non-command interfaces, *Advances in Human-Computer Interaction*, Ablex Publishing Corporation, Norwood, New Jersey, vol. 4, chapter 6, pp. 151-190.

[Kim *et al.* 2011] Kim, J.I., Park, I.W., Lee, H.H., An Intelligent Context-Aware Learning System Based on Mobile Augmented Reality, *International conference on Ubiquitous Computing and Multimedia Applications*, pp. 255-264, Daejeon, Korea, 2011.

[Klein and Murray 2007] Klein, G., Murray, D., 2007. Parallel Tracking and Mapping for Small AR Workspaces, *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 225 – 234.

[Knopfle *et al.* 2005] Knopfle, C., Weidenhausen, J., Chauvigne, L., Stock, I., 2005. Template Based Authoring for AR Based Service Scenarios, *IEEE Conference on Virtual Reality*, Bonn, Germany, pp. 249-252.

[Kuzuoka 1992] Kuzuoka, H., 1992. Spatial workspace collaboration: A shared view video support system for remote collaboration capability, *the SIGCHI conference on Human factors in computing systems*, pp. 533–540.

[Ledermann and Schmalstieg 2005] Ledermann, F., Schmalstieg, D., 2005. APRIL: a high-level framework for creating augmented reality presentations, *IEEE Virtual Reality*, Bonn, pp. 187-194

[Lee and Akin 2011] Lee, S., Akin, O., 2011, Augmented reality-based computational fieldwork support for equipment operations and maintenance, *Automation in Construction*, vol. 20, no. 4, pp. 338–352

[Lee and Kim 2009] Lee, G.A., Kim, G.J., 2009. Immersive authoring of tangible augmented reality content: a user study, *Journal of Visual Languages and Computing*, vol. 20, no. 2, pp. 61-79.

[Lee and Rhee 2008] Lee, J.Y., Rhee, G., Context-aware 3D visualization and collaboration services for ubiquitous cars using augmented reality, *The International Journal of Advanced Manufacturing Technology*, vol. 37, no. 5-6. pp. 431-442, 2008.

[Lee *et al.* 2008] Lee, J.Y., Seo, D.W., Rhee, G., Visualization and interaction of pervasive services using context-aware augmented reality, *Expert Systems with Applications*, vol. 35, no. 4, pp. 1873-1882, 2008.

[Lee *et al.* 2011] Lee, J.Y., Seo, D.W., Rhee, G.W., 2011. Tangible authoring of 3D virtual scenes in dynamic augmented reality environment, Computers in Industry, vol. 62, no. 1, pp. 107-19.

[Looser *et al.* 2006] Looser, J., Grasset, R., Seichter, H., Billinghurst, M., 2006. OSGART - A Pragmatic Approach to MR, *IEEE and ACM International Symposium on Mixed and Augmented Reality*, Santa Barbara, CA, USA, pp. 22-25

[MacIntyre *et al.* 2004] MacIntyre, B., Gandy, M., Dow, S., Bolter, J., 2004. DART: a toolkit for rapid design exploration of augmented reality experiences, ACM Symposium on User Interface Software and Technology, New York, USA, pp.197–206.

[Milgram and Kishino 1994] Milgram, P., and Kishino, F., 1994. A taxonomy of mixed reality virtual display. *Institute of Electronics, Information and Communication Engineers (IEICE) Transactions on Information Systems*, vol. E77-D, no. 9, pp. 1321-1329.

[Milgram *et al.* 1994] Milgram, P., Takemura, H., Utsumi, A., and Kishino, F., 1994. Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum, *Proceedings of Telemanipulator and Telepresence Technologies*, Hari Das, pp. 282-292.

[Nee *et al.* 2012] Nee, A.Y.C., Ong, S.K., Chryssolouris, G., Mourtzis, D., 2012, Augmented reality applications in design and manufacturing, *Annals of CIRP*, vol. 61, no. 2, pp. 657–679

[Neubert *et al.* 2012] Neubert, J., Pretlove, J., Drummond, T., 2012, Rapidly constructed appearance models for tracking in augmented reality applications, *Machine Vision and Applications*, vol. 23, no. 5, pp. 843-856

[OpenCV] Open Source Computer Vision Library, http://sourceforge.net/projects/opencvlibrary/, [last access date: 24th, September, 2012].

[OpenSceneGraph] OpenSceneGraph, http://www.openscenegraph.org/projects/osg [last access date: 24th, September, 2012].

[OWL API] The OWL API, http://owlapi.sourceforge.net/, [last access date: 24th, September, 2012].

[Park *et al.* 2008] Park, H.M., Lee, S.H., Choi, J.S., 2008. Wearable Augmented Reality System using Gaze Interaction, *IEEE International Symposium on Mixed and Augmented Reality*, Cambridge, UK, pp. 175-176.

[Pellet] Pellet: OWL 2 Reasoner for Java, http://clarkparsia.com/pellet, [last access date: 24th, September, 2012].

[Reitmayr *et al.* 2007] Reitmayr, G., Eade, E., and Drummond, T.W. 2007, Semi-automatic Annotations in Unknown Environment, *IEEE and ACM International Symposium on Mixed and Augmented Reality*, Nara, Japan, pp. 1-4.

[Rekimoto 1995] Rekimoto, J., 1995. The Magnifying Glass Approach to Augmented Reality Systems, *International Conference on Artificial Reality and Tele-Existence / Conference on Virtual Reality Software and Technology*, Makuhari, Japan, pp. 123-132.

[Sakata *et al.* 2006] Sakata, N., Kurata, T., Kuzuoka, H., 2006. Visual Assist with a Laser Pointer and Wearable Display for Remote Collaboration, *Transactions of the Virtual Reality Society of Japan*, vol. 11, vo. 4, pp. 561-568.

[Savioja *et al.* 2007] Savioja, P., Järvinen, P., Karhela, T., Siltanen, P., Woodward, C., 2007, Developing a mobile, service-based augmented reality tool for modern maintenance work, *the 2nd international conference on Virtual Reality*, Beijing, China, pp 554–563

[Schwald *et al.* 2001] Schwald. B., Figue, J., Chauvineau, E., Hong, F.V., Robert, A., Arbolino, M., Schnaider, M., de Laval B, de Rauly, F.D., Anez, F.G., Baldo, O., Santos, J., 2001, STARMATE: using augmented reality technology for computer guided maintenance of complex mechanical elements, *eBusiness and eWork Conference*, Venice-Italy, pp 17–19

[Setchi and White 2003] Setchi, R., White, D., 2003. The development of a hypermedia maintenance manual for an advanced manufacturing company. *International Journal of Advanced Manufacturing Technology*, vol. 22, no. 5–6, pp. 456–464.

[Shen *et al.* 2011] Shen Y., Ong S.K., Nee A.Y.C., 2011. Vision-based hand interaction in augmented reality environment. International Journal of Human-Computer Interaction, vol. 27, no. 6, pp. 523–544.

[Shin *et al.* 2009] Shin, C., Lee, W., Suh, Y., Yoon, H., Lee, Y., Woo, W., CAMAR 2.0: Future Direction of Context-Aware Mobile Augmented Reality, *International Symposium on Ubiquitous Virtual Reality*, pp. 21-24, Gwangju, 2009.

[Sun *et al.* 2007] Sun, M., Li, M., Zhang, F., Wang, Z., Wu, D., 2007. Hybrid tracking for augmented reality GIS registration, *Japan-China Joint Workshop on Frontier of Computer Science and Technology*, pp. 139-145.

[SWRL] SWRL: A Semantic Web Rule Language Combining OWL and RuleML, http://www.w3.org/Submission/SWRL/, [last access date: 24th, September, 2012].

[Venezia and Marengo 2010] Venezia, C., Marengo, M., Context Awareness Aims at Novel Fruition Models: Augmented Reality May be the Killer Application for Context Awareness, *IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing*, pp. 392-396, Hong Kong, 2010.

[Wang *et al.* 2004] Wang, X.H., Zhang, D., Gu, T., Pung, H.H., 2004. Ontology-based context modeling and reasoning using OWL, *2nd IEEE Annual* *conference on Pervasive Computing and Communications Workshops*, Singapore, pp. 18-22.

[Williams et al. 2007] Williams B., Klein G., Reid I., 2001. Real-time SLAM relocalisation, 11th IEEE International Conference on Computer Vision (ICCV’07), Rio de Janeiro, October 2007, pp. 1-8.

[Yan *et al.* 2009] Yan, W., Ishii, H., Shimoda, H., Izumi, M., A Feasible Tracking Method of Augmented Reality for Supporting Fieldwork of Nuclear Power Plant, *Lecture Notes in Computer Science*, vol. 5622 LNCS, pp. 639-646, 2009

[Yuan *et al.* 2004] Yuan, M.L., Ong, S.K., Nee, A.Y.C., 2004. The Virtual Interaction Panel: An Easy Control Tool in Augmented Reality Systems, *Computer Animation and Virtual Worlds Journal, Special Issue: The Very Best Papers from CASA 2004*, vol. 15, no. 3-4, pp. 425-432.

[Zhang *et al.* 2010] Zhang, J.X., Sheng, Y.H., Hao, W., Wang, P.P., Tian, P., Miao, K., Pickering, C.K., A Context-aware Framework Supporting Complex Ubiquitous Scenarios with Augmented Reality Enabled, *International Conference on Pervasive Computing and Applications*, pp. 69-74, Maribor, 2010

[Zhou *et al.* 2008] Zhou, F., Duh, H.B.-L., and Billinghurst, M., 2008. Trends in augmented reality tracking, interaction and display: a review of ten years of ISMAR, *IEEE and ACM International Symposium on Mixed and Augmented Reality*, Cambridge UK, pp. 193-202.

Appendix A Survey Questionnaire on ARAMS Usability in Remote Maintenance

Please complete the following information identifying the person completing this part of the Statistical Report. This will help if questions arise in interpreting the data.

Personal information

Name:	______________________________
Gender:	______________________________
Age:	______________________________
Occupation:	______________________________

Contact information

Phone Number:	______________________________
Email Address:	______________________________
Date:	______________________________

Remarks

We appreciate your time and effort for this case study. The data received will be collected but *NO PERSONAL* data will be released without your specific consent.

Part I

Background knowledge

1. Do you have any experience of online remote collaboration using conventional tools, e.g., phones and Skype? (_____)

A. Yes, and often

B. Yes, but seldom

C. No

D. ________________ (Please specify the name of the collaborative tool if Yes)

2. Describe your knowledge of Augmented Reality technology. (_____)

A. Expert

B. Intermediate

C. Beginner

D. Unknown

3. Describe your knowledge of desktop maintenance. (_____)

A. Expert

B. Intermediate

C. Beginner

D. Unknown

Part II

Case study

Table A.1: Quantitative data collection

Parameter / Step	Method (Traditional method / Proposed method)	
	Completion time /(s)	Number of errors
Open the computer chassis		
Unfasten the spring		
Slide out the plate		
Unplug the wires		
Slide out the CD tray		
Total		

Part III

Post-case study

1. Please evaluate the intuitiveness, satisfaction level, and ease of use of the method you have used in the case study (1=most negative, 5=most positive).

Table A.2: Qualitative analysis

Method / Parameter	Role (Expert / Technician)	
	Traditional method	Proposed method
Intuitiveness		
Satisfaction		
Ease of use		

2. Please provide any additional comments or suggestions on proposed AR-assisted remote maintenance system.

Appendix B Survey Questionnaire on ARAMS Usability in Context-aware AR Contents Authoring

Please complete the following information identifying the person completing this part of the Statistical Report. This will help if questions arise in interpreting the data.

Personal information

Name: ____________________

Gender: ____________________

Age: ____________________

Occupation: ____________________

Contact information

Phone Number: ____________________

Email Address: ____________________

Date: ____________________

Remarks

We appreciate your time and effort for this case study. The data received will be collected but *NO PERSONAL* data will be released without your specific consent.

Part I
Background knowledge

1. Do you have any experience of developing Augmented Reality applications? (_____)

A. Yes, and often

B. Yes, but seldom

C. No

D. ________________ (Please specify the name of the collaborative tool if Yes)

2. Describe your knowledge of Augmented Reality technology. (_____)

A. Expert

B. Intermediate

C. Beginner

D. Unknown

3. Describe your knowledge of context-awareness. (_____)

A. Expert

B. Intermediate

C. Beginner

D. Unknown

Part II

Case study

Table B.1: Task completion collection

Authoring System	ARAMS	BuildAR	iaTAR
Completion time			

Part III

Post-case study

1. Please evaluate the three authoring systems in terms of ease of learning and ease of use (1=most negative, 5=most positive). In addition, please provide feedback on the usability of the three systems.

Table B.2: Qualitative analysis

		ARAMS	BuildAR	iaTAR
Ease of learning				
Ease of use				
Usability	Pros			
	Cons			

2. Please provide any additional comments or suggestions on proposed authoring system.

Appendix C Survey Questionnaire on ARAMS Usability in Context-aware AR Contents Visualization

Please complete the following information identifying the person completing this part of the Statistical Report. This will help if questions arise in interpreting the data.

Personal information

Name: ______________________________

Gender: ______________________________

Age: ______________________________

Occupation: ______________________________

Contact information

Phone Number: ______________________________

Email Address: ______________________________

Date: ______________________________

Remarks

We appreciate your time and effort for this case study. The data received will be collected but *NO PERSONAL* data will be released without your specific consent.

Part I

Background knowledge

1. Do you have any experience of CNC machine maintenance using conventional maintenance manuals, e.g., paper prints or mobile devices? (_____)

A. Yes, and often

B. Yes, but seldom

C. No

D. ________________ (Please specify the type of maintenance manual if Yes)

2. Describe your knowledge of Augmented Reality technology. (_____)

A. Expert

B. Intermediate

C. Beginner

D. Unknown

3. Describe your knowledge of context-awareness. (_____)

A. Expert

B. Intermediate

C. Beginner

D. Unknown

Part II

Case study

Table C.1: Quantitative data collection

Step / Parameter			FuseBlownOut	Relay Loose	PCB Defective	Total
Group (Group I/ Group II)	Completion time /(s)	First time				
		Second time				

Part III

Post-case study

1. Please evaluate the intuitiveness, satisfaction level, and ease of use of the maintenance supporting tools you have used in the case study (1=most negative, 5=most positive).

Table C.2: Qualitative analysis

Step / Parameter	Intuitiveness	Satisfaction	Ease of use
Paper-based Manuals			
Traditional AR-assisted maintenance system			
Proposed system			

2. Please provide any additional comments or suggestions on proposed AR-assisted maintenance system.